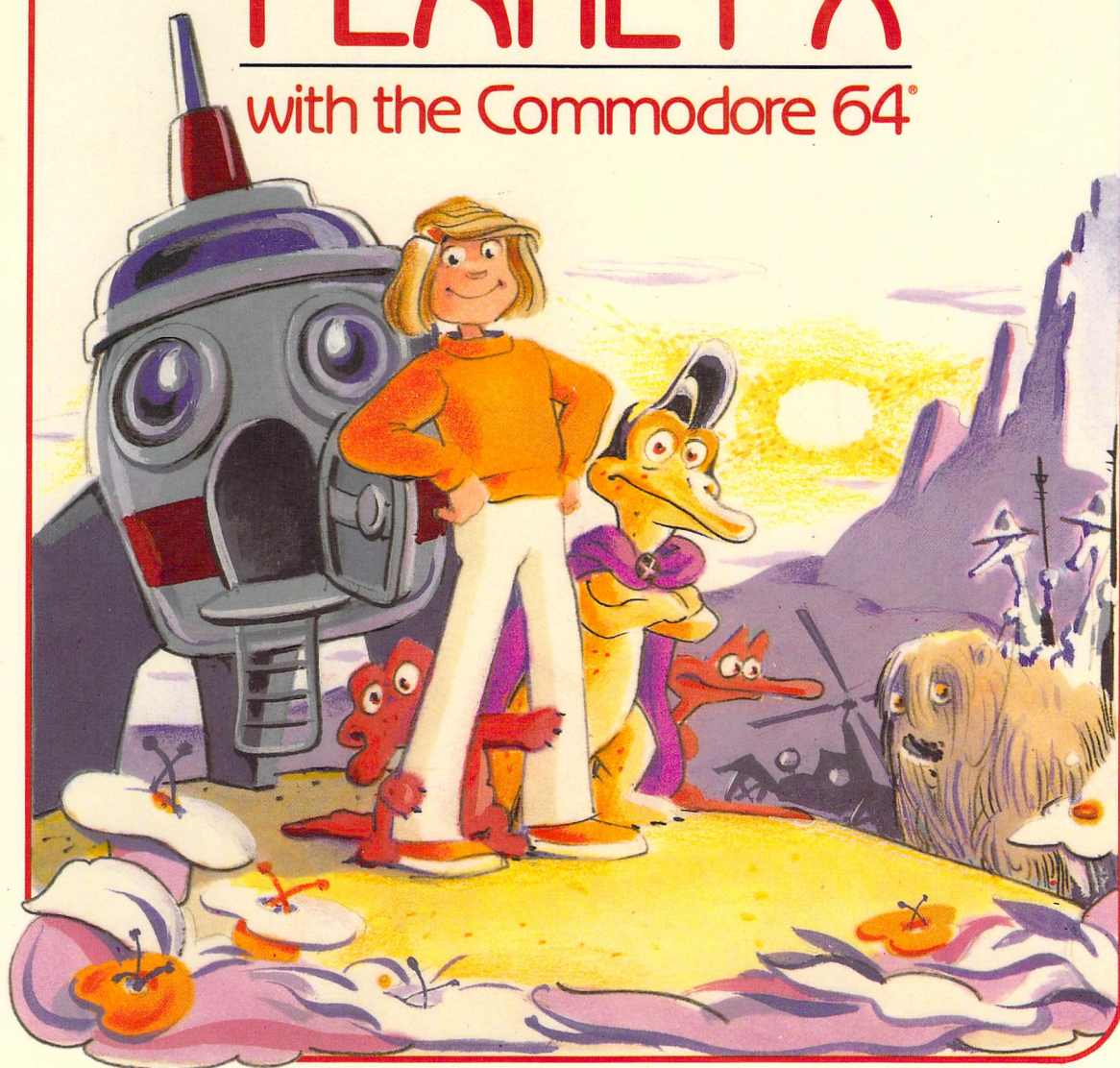


Michael Orkin Ed Bogas

SURVIVAL on PLANET X

with the Commodore 64®



SURVIVAL ON PLANET X

SURVIVAL ON PLANET X

WITH THE COMMODORE 64®

**Michael Orkin
Ed Bogas**

Illustrated by William Luckey



**A Reston Computer Group Book
Reston Publishing Company, Inc.
A Prentice-Hall Company
Reston, Virginia**

Library of Congress Cataloging in Publication Data

Orkin, Michael.

Survival on Planet X with the Commodore 64.

(A Reston computer group book)

Includes index.

1. Commodore 64 (Computer) — Programming. 2. Basic
(Computer program language) I. Bogas, Ed. II. Title.

QA76.8.C64074 1984 001.64'2 83-24508

ISBN 0-8359-7413-8

Copyright© 1984 by Reston Publishing Company, Inc.,
A Prentice-Hall Company,
Reston, Virginia 22090

All rights reserved. No part of this book may be reproduced in any way, or
by any means, without written permission from the publisher.

The final typeset pages were produced on a TyXSET 1000 system in Reston,
Virginia, using a Mergenthaler Omnitech/2100. The galley and page proofs
were produced on the TyXSET 1000 system using a Canon LBP-10 Laser
Printer.

TyXSET 1000 is a trademark of TyX Corporation
Commodore 64 is a registered trademark of Commodore Business Machines

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

CONTENTS

Preface	vii
Chapter 1: A CALL FOR HELP	1
Chapter 2: XANG	9
Chapter 3: THE LEADERSHIP COUNCIL	21
Chapter 4: ESCAPE	31
Chapter 5: FLIGHT TO THE HILLS	49
Chapter 6: XAVIAR	63
Chapter 7: THE TROLLEY	75
Chapter 8: THE ARMY LIZARDS	87
Chapter 9: BACK TO THE SPACESHIP	101
Chapter 10: ANOTHER CALL FOR HELP	123
Glossary of Programming Terms	141
Index of Programming Terms	163

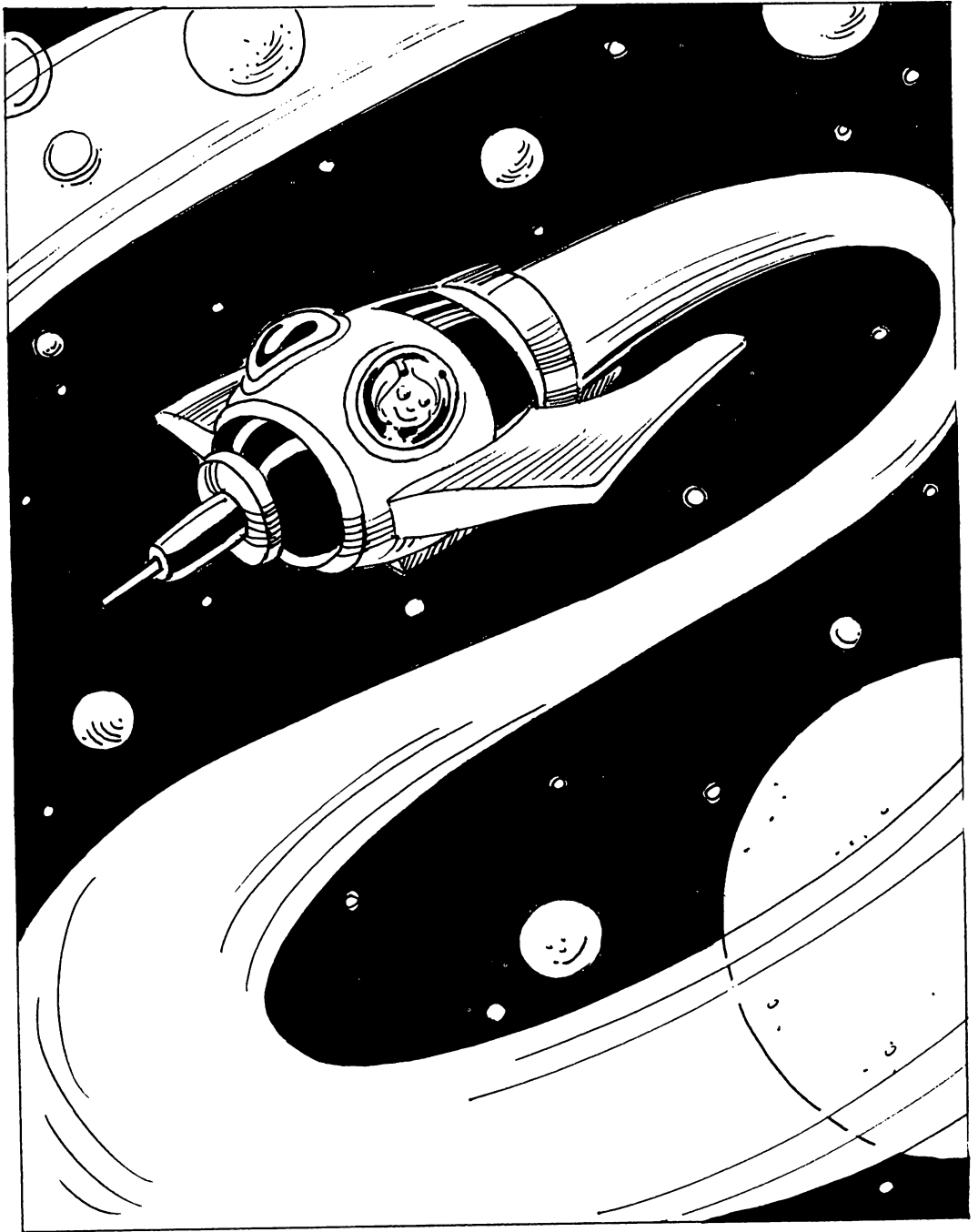
PREFACE

Survival on Planet X is an adventure story that includes computer programs illustrating events in the story. As designers of video games, we have tried to present the basic ideas of programming in an exciting way. The programs are written in the BASIC programming language and can be used on a Commodore 64® home computer. You don't have to know anything about computers to use the programs. In fact, you can read the story without using a computer at all.

Some of the programs in this book are easy to understand and some are complicated. All are short enough to easily type into your computer. In the back of the book is a glossary of programming procedures that you can use as a guide.

Computers are not as smart as people and can only understand precise instructions. When you use the programs in the book, you must be sure to type them into the computer exactly as they are printed. Even a missing comma or misplaced quotation mark can hopelessly confuse the computer.

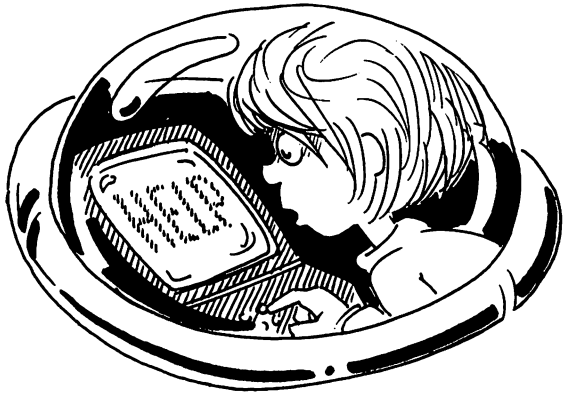
Programs used for video games, business, or scientific applications may be longer and more complicated than the programs in the book, but the ideas are the same. We hope you enjoy *Survival on Planet X* and have fun experimenting with your computer.



A CALL FOR HELP

Not wanting to spend another dull summer babysitting and visiting relatives, Vivian decided to take her spaceship on a trip through an uncharted solar system. She enjoyed being alone and was able to relax in the silence of outer space.

One afternoon (at least it was afternoon somewhere on Earth), just after Vivian put her ship on automatic control in order to watch a meteor storm, she noticed the message "HELP!" on her computer communications screen. She immediately used her computer to trace the message and discovered that it came from a small, unexplored planet named Planet X. Vivian decided to investigate and changed course. Soon she was heading toward the small planet.



We will use the Commodore 64 home computer to help describe Vivian's adventure. As a result, you will learn about computer programming. The programming language we will use is called Commodore 64 BASIC, which is built into the Commodore 64. When you turn on your computer, "READY" will appear in the upper left-hand corner. You can get the computer to print "HELP" by typing

```
PRINT "HELP"
```

and pressing the **RETURN** key. Try it.

To get the computer to repeatedly print "HELP" the way it appeared on Vivian's computer, we need a program. A computer program is a set of instructions that tells the computer what to do.

Here are some guidelines you can refer to throughout the story. Whenever you finish typing a line in a program, you must press the **RETURN** key before you start the next line. To correct a mistake before you are finished typing a line, use the **INST/DEL** key which backspaces and erases characters. (Find this key on your keyboard.) To correct an already typed line, you can retype it. The computer will automatically put it in its proper place.

If a program has a mistake in it that you can't find, you can erase the program from the computer's memory and start over. You do this by typing NEW and pressing the **RETURN** key or by turning the computer off and turning it on again. To clear the screen without erasing the computer's memory, hold down the **SHIFT** key and press the **CLR/HOME** key.

Now type the following program. Press **RETURN** after each line.

```
10 PRINT "HELP! "  
20 GOTO 10
```

This program puts the computer in an unending "loop." Once it starts, you can stop it by pressing the **RUN/STOP** key. To start the program, type RUN and press **RETURN**. Do this now.

Line 20 in the above program is called a *GOTO statement*. It instructs the computer to go to another line in the program, in this case line 10. Therefore, the computer keeps printing HELP until you stop it. Type LIST and press **RETURN**. The computer will print a list of the program. Now change line 10 by adding a semicolon after the quotation mark. Type

```
10 PRINT "HELP! ";
```

RUN this program. The semicolon at the end of a PRINT statement tells the computer to carry out the next PRINT statement at the next available space. If there is no punctuation mark, as before, the computer skips to the next line before printing again. You can slow down the display by holding down the **CTRL** (control) key as the program is running.

Now change line 10 once more by typing a comma instead of a semicolon at the end of the PRINT statement. Type

```
10 PRINT "HELP! ",
```

Now RUN the program.

After two long days of travel, Vivian reached Planet X. She landed her spaceship in a valley surrounded by large, rocky mountains. Vivian's computer informed her that she could breathe the air. Emerging from her spaceship, she met a strange creature who looked something like a lizard but stood upright. Vivian was a bit scared but reassured herself that she could at least understand the lizard creature with the help of the translating system on her computer.



"My name is Xang," said the creature who greeted Vivian. "My job is to welcome visitors to Planet X. Welcome to Planet X! There are very few visitors to Planet X. In fact, I've never welcomed anyone before. I usually sit in my office all day and wait for someone to arrive. You can't imagine how excited I am."

Type NEW to erase the current program from the computer's memory. We will write a new program that shows Xang's excitement. Enter the following program:

```
10 PRINT "A VISITOR! "  
20 PRINT  
30 PRINT "I CAN'T BELIEVE IT! "  
40 PRINT  
50 GOTO 10
```

RUN this program. The PRINT statement alone tells the computer to skip a line. Now type the following lines. They cause a delay in Xang's speech. We will call the statements *delay statements*. The computer will automatically put them in the right sequence.

```
25 FOR N = 1 TO 500: NEXT N
45 FOR N = 1 TO 500: NEXT N
```

RUN this program. Xang seems to have calmed down a little. Now LIST the program. Notice that the statements are in the right sequence. It is a good idea to number statements at least by 10's so you will have room to insert new statements.

Vivian had never seen a lizard creature before and was happy to see that he was friendly. She wondered who had sent the HELP message.

XANG

After Xang welcomed Vivian to Planet X, he handed her a form and said, "You must fill out a Planet X visitor's card which all visitors to Planet X are required to complete."

Here are some programs that show what a Planet X visitor's card looks like. The first program prints a blank form with a heading. To clear the computer's memory, type NEW and press **RETURN**. Then type the following program and run it:

```
10 PRINT "PLANET X VISITOR'S CARD"
20 FOR N = 1 TO 21
30 PRINT
40 NEXT N
```

Lines 20 through 40 in the above program are called a *FOR/NEXT* loop. After the computer reads line 20, it performs every instruction until the NEXT statement and then returns to line 20.

Every time it does this, the computer increases the value of N by 1 until N equals 21. In this way, the letter N acts as a "counting variable." You can use any letter for the counting variable. When the loop has been executed



21 times, the computer moves on to the next instruction if there is one. In this program, the only instruction in the FOR/NEXT loop is the PRINT statement in line 30. Since a blank PRINT statement causes the computer to skip a line, the loop causes the computer to skip 21 lines. In the last chapter, we used a FOR/NEXT loop with no instructions between the FOR and NEXT statements:

```
FOR N = 1 TO 500: NEXT N
```

The computer went through the loop 500 times without doing anything. The time it took the computer to do this caused the delay between Xang's exclamations. During each brief pause, the computer was actually reading the FOR/NEXT instructions 500 times. In Commodore 64 BASIC, you can put more than one instruction on a single line if you separate each instruction by a colon. We did this for the delay statement.

Now we will change the Visitor's Card. Type the following lines and RUN the program:

```
12 PRINT  
14 PRINT "VISITOR#", "NAME", "HOME"  
20 FOR N = 1 TO 19  
30 PRINT N, ".....", "....."
```

When a comma occurs in a PRINT statement, it skips to the next position, like a tab key on a typewriter. In Commodore 64 BASIC, the tab settings are every 10 spaces across a line. Since the screen is 40 spaces wide, the comma in the PRINT statement makes it easy to write a program that prints information in four equal columns. In the last chapter, the comma was used in a PRINT statement to make four columns of HELP messages. This Visitor's Card numbers the lines that visitors are supposed to sign and has dotted lines to show where they are supposed to sign it. LIST the program. It should look like this:

```
10 PRINT "PLANET X VISITOR'S CARD"
12 PRINT
14 PRINT "VISITOR#", "NAME", "HOME"
20 FOR N = 1 TO 19
30 PRINT N, ".....", "....."
40 NEXT N
```

The Visitor's Card Xang handed Vivian was blank. "You might be the first one who has ever visited Planet X," he said. "Here is a demonstration card you can look at in case you need help filling out the card."

To see the demonstration card, change line 30 as follows and RUN the revised program.

```
30 PRINT N, "XANG", "PLANET X"
```

Vivian completed the Visitor's Card and Xang asked her to pronounce her name. He told her, "I know a poem about someone named Vivian" and recited a poem with her name in it.

Information such as your name is stored in the computer's memory the way you would put something in a box and label the box so you know where to find it. This storage location with a label is called a *variable*. If you are storing nonnumeric information, the variable is called a *string variable*. A string variable can be denoted by a letter followed by a dollar sign, like A\$ or B\$, or a letter followed by a number, like A2\$, or two letters, like AB\$. In Commodore 64 BASIC there is, however, a special string TI\$, which cannot be used like the others. (We will explain why later.)

One method of putting information into the computer's memory is by using an INPUT statement. When the computer reads an INPUT statement, it will print a question mark on the screen and wait until you type in the information you want to input. When you do this and press **RETURN**, the computer will proceed to the next instruction.

Here is a program that uses the INPUT statement. Type NEW before entering the program so the current program will be erased. Then enter the program and RUN it.

```
10 PRINT "WHAT IS YOUR NAME? "  
20 PRINT  
30 INPUT A$  
40 PRINT A$; "? THAT'S FUNNY. "  
50 PRINT "I KNOW A POEM ABOUT"  
60 PRINT "SOMEONE NAMED "; A$; ". "
```

Notice that in lines 40 and 60, A\$ is not included in quotes. If it were, the computer would print "A\$" rather than the current value of the variable A\$, namely, your name.

Now we will instruct the computer to print the poem with your name. Add the following lines and RUN the revised program.

```
70 PRINT
80 PRINT "THERE ONCE WAS A RULER"
90 PRINT "NAMED "; A$; " THE THIRD"
100 PRINT "WHO RULED A LARGE KINGDOM"
110 PRINT "WITHOUT SAYING A WORD"
```

If you have trouble getting the program to run, LIST it and check to see that it is correctly typed. Even a forgotten or incorrect punctuation mark could keep a program from running. Now complete the poem by adding the following lines. Then RUN the program.

```
120 PRINT "WHEN THE PEOPLE CRIED "; A$
130 PRINT "LET YOUR RULINGS BE HEARD"
140 PRINT "OFF WITH YOUR HEADS! "
150 PRINT "SAID "; A$; " THE THIRD. "
```

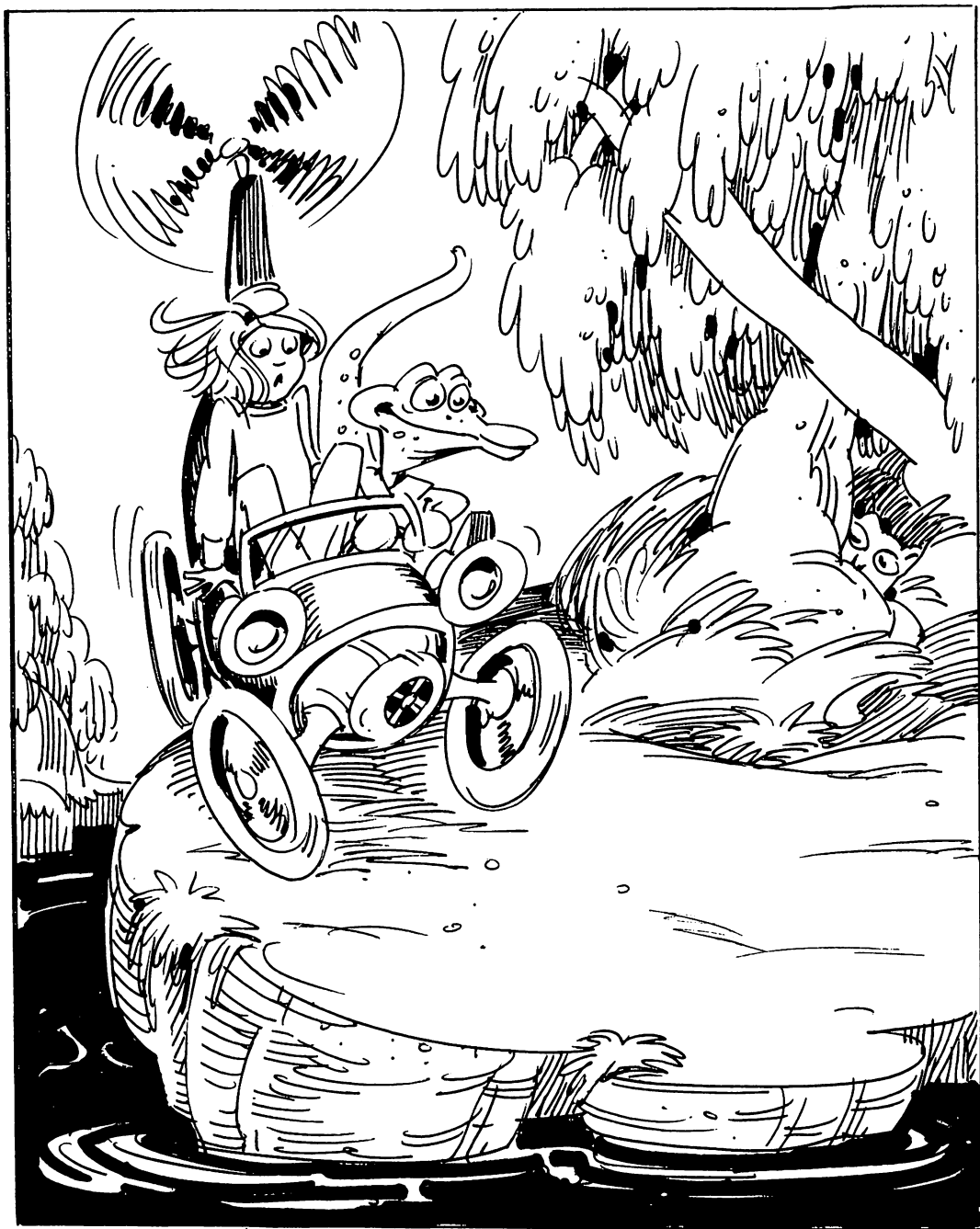
RUN this program a few times, inputting a different weird name each time.

Vivian was pleased to hear a poem about herself and told Xang, "I really enjoyed the poem, but I'm here to investigate a signal for help that I received in space." She asked him if he knew anything about it. Xang replied that his job was to greet visitors to Planet X and that he didn't really know much about anything else.

"The Leadership Council would know about things like signals for help," said Xang. "I will take you to see them."

He walked over to a vehicle that looked like a small car with a propeller on the back. He climbed into the car and motioned for Vivian to sit next to him. She did this with some difficulty, since the vehicle was built for lizard creatures, not Earthlings. Xang started the propeller car, which made a loud whirring sound, and he and Vivian sped off down a bumpy road.

As she and Xang traveled, Vivian was able to see the countryside of Planet X. A bright sun shone in the Planet X sky. Tall, greenish-purple trees that looked like painted weeping willows were scattered along the dirt road. A stream with purple water flowed along the roadside. A pair of strange animals that resembled giant hamsters walked slowly along the bank of the stream. Xang told Vivian that these animals were called xamsters. Thin green birds with huge eyes and long beaks flew by overhead. A small blue animal like a cat darted behind a tree. Some lizard creatures who looked like Xang were walking along the side of the road carrying what appeared to be garden tools.





One of the creatures was holding a bushy purple plant. As Vivian and Xang drove by, the creatures stared straight ahead with blank expressions.

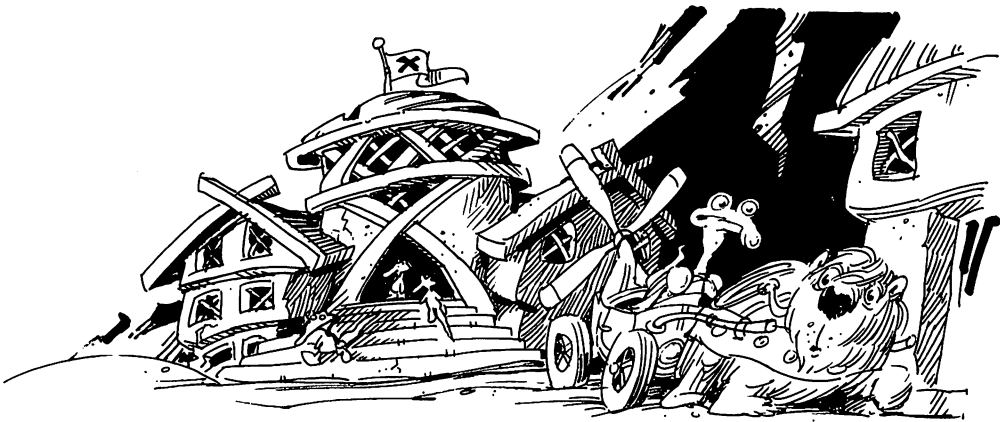
Further on, the travelers passed another propeller car. There were about six lizard creatures in it. Its propeller wasn't spinning; it was being pulled by a xamster.

Abandoned in the middle of a field, Vivian saw a piece of farm machinery that looked like a tractor. The tractor had not



been used for some time and was missing all of its wheels. Plants were growing from it and it was covered with purple rust. A few lizard creatures were sitting around a small fire next to the tractor. They appeared to be cooking something. In the distance there was a rundown house. A propeller car with the front end smashed in was in a dirt driveway next to the house.

After about an hour's travel, Vivian and Xang came to a city located at the base of a mountain. A market in the cen-



ter of town was crowded with lizard creatures carrying baskets and pulling carts. Larger carts and propeller cars were being pulled through the streets by xamsters. Other strange creatures wandered through the streets among the lizard creatures. Finally, Xang parked the propeller car next to a large stone building that seemed on the verge of collapse.

“This is government headquarters,” said Xang proudly.
“The Leadership Council is here.”

THE LEADERSHIP COUNCIL

Vivian and Xang entered government headquarters and were met at the door by an armed lizard guard. Xang explained that Vivian was a visitor to Planet X and that she wanted to see the Leadership Council. The guard rudely informed Xang that no visitor to Planet X had ever been allowed to see the Leadership Council. Vivian pointed out that the reason no visitors had ever been allowed to see the Leadership Council was that there had never been any visitors to Planet X before.

The guard looked at Vivian suspiciously and told her that he did not have the authority to grant her request and that, if she wanted to discuss the matter further, she would have to speak with Head Administrator #1 whose office was down the hall and to the right. "I guess that's what we'll have to do," said Vivian, who turned and walked down the hall with Xang following close behind.

Vivian and Xang entered the office of Head Administrator #1. Xang greeted the administrator and said, "I have brought a visitor who requests to see the Leadership Council."

"No visitor has ever been allowed to see the Leadership Council," said Head Administrator #1. "I don't have the authority to grant such a request. Try Head Administrator #2. His office is down the hall and to the right."

"This is worse than school," said Vivian, who hated it when people couldn't make decisions. With Xang close behind, she

marched down the hall and to the right to the office of Head Administrator #2. Unfortunately, Head Administrator #2 gave the same response as Head Administrator #1. Before long, Vivian and Xang found they were in the office of Head Administrator #3. After getting the same response there, Vivian and Xang went to the office of Head Administrator #4, who proceeded to send them down the hall and to the right and back to the office of Head Administrator #1. After walking around the halls until they were quite dizzy, Vivian and Xang reached the office of Head Administrator #4 for the sixth time.

Type and RUN the following program;it will demonstrate Vivian's difficulty in trying to find an administrator who would grant her request to see the Leadership Council. Clear the computer's memory before starting. We will explain the POKE statements later. (They are used here to make the sound of footsteps.)

```
20 FOR X = 1 TO 4
25 POKE 54296, 15
27 POKE 54273, 80 : POKE 54272, 80
30 PRINT "HI, I'M ADMINISTRATOR"
40 PRINT "NUMBER"; X
```

```
50 FOR L = 1 TO 200: NEXT L
70 FOR W = 1 TO 25
80 PRINT "WALK";
90 POKE 54276,129
100 FOR L = 1 TO 50: NEXT L
120 POKE 54276, 128
130 FOR L = 1 TO 50: NEXT L
150 NEXT W
160 PRINT: PRINT
170 NEXT X
190 POKE 54296,0
```

List the program and try to understand its logic. Notice that lines 50, 100, and 130 are delay statements. To visit each of the four administrators six times, add these lines and RUN the program.

```
10 FOR K = 1 TO 6
180 NEXT K
```

When Vivian and Xang reached the office of Head Administrator #4 for the sixth time, she told him that although it was true that no visitor to Planet X had ever been given permission to see the Leadership Council, it was



also true that no visitor had ever been refused permission to see the Leadership Council. If the administrator refused to grant her permission, he would be the first one to ever refuse to allow a visitor to see the Leadership Council.

After thinking about this, Head Administrator #4 said, "I think you may be correct and I grant your request. The Leadership Council is upstairs and down the hall."

Vivian and Xang went upstairs and down the hall and entered the office of the Leadership Council. A lizard creature with long lizard hair and big lizard eyes greeted them. "My name is Xanada," said the lizard creature. "It is my job to make sure that everyone who sees the Leadership Council is intelligent. Therefore, I will give you a test. There are five questions. If you give a wrong answer you must go to jail. First question: How much is 36,287 plus 48,898?"

"Wait a minute," said Vivian. "I only came here to answer a signal for help. I don't want to go to jail. I think I'll just leave now." Vivian turned to go, but was stopped by two armed lizard guards. Xang was nowhere to be seen.

"I'm sorry," said Xanada, "but you can't leave now."

Realizing that she had no choice, Vivian used her pocket computer to quickly solve the first problem.

In BASIC you can do arithmetic by using the PRINT statement. The symbol for addition is "+"; for subtrac-

tion it is “-”; for multiplication it is “*”; and for division it is “/”. For example, to add 8 plus 9, just type

```
PRINT 8 + 9
```

and press **RETURN**. To multiply 17 times 12, just type

```
PRINT 17 * 12
```

and press **RETURN**. Experiment with the PRINT statement to do more arithmetic problems.

We will write a program that instructs the computer to add any two numbers. Just as string variables are used to input words, numeric variables are used to input numbers. They can be denoted by letters like X or Y, letters followed by numbers, like X1 or Y2, or pairs of letters like AB. (The variable names ST and TI are reserved for special use.) You don’t need a DIM statement with numeric variables. Clear the computer’s memory. Enter the following program and RUN it several times, each time adding two different numbers. The GOTO statement keeps the program running. To stop the program, hold down the **RUN/STOP** key and press **RESTORE**.

```
10 PRINT "THIS PROGRAM ADDS TWO NUMBERS"  
20 PRINT "INPUT FIRST NUMBER"  
30 INPUT X  
40 PRINT "INPUT SECOND NUMBER"
```

```
50 INPUT Y
60 PRINT
70 PRINT "THE ANSWER IS "; X + Y
80 PRINT
90 GOTO 20
```

Vivian correctly answered the first problem and Xanada gave her the second problem: "How much is 87,456 minus 63,084?"

The addition program can be easily changed to a subtraction program. Make the following changes and RUN the subtraction program. Use it to answer Xanada's second question.

```
10 PRINT "THIS PROGRAM SUBTRACTS THE SECOND"
15 PRINT "NUMBER FROM THE FIRST"
70 PRINT "THE ANSWER IS "; X - Y
```

After Vivian answered the second problem correctly, Xanada gave her the third problem: "How much is 2,867 times 1,923?"

Now we will change the subtraction program to a multiplication program. Sometimes, when numbers are too large or too small to print in the usual way, the computer prints them in a peculiar looking form called *exponential notation*. To convert a number in exponential notation to a number in ordinary form, you move the decimal point over as many spaces as are indicated after the E. For example, 2.31E4 indicates that the decimal point should be moved four places to the right: 23,100. The notation 81.22E-5 means to move the decimal point five places to the left: .0008122. Make the following changes and RUN the resulting program. Use it to answer Xanada's third question.

Type 15 and press **RETURN** to erase line 15. Then type

```
10 PRINT "THIS PROGRAM MULTIPLIES TWO NUMBERS"  
70 PRINT "THE ANSWER IS "; X * Y
```

List the program and experiment with it until you understand how it works.

Vivian answered the third problem and Xanada gave her the fourth one: "How much is 619,203 divided by 39?"

Make the indicated changes and RUN this program to answer Xanada's fourth question.

```
10 PRINT "THIS PROGRAM DIVIDES THE FIRST NUMBER"  
15 PRINT "BY THE SECOND"  
70 PRINT "THE ANSWER IS "; X/Y
```

When Vivian answered the fourth question correctly Xanada said, "Here is the fifth question: Who was the first ruler of Planet X?"

"How can I answer that?" exclaimed Vivian. "I never studied the history of Planet X!"

"Enough of this insolence!" snapped Xanada. She turned to the lizard guards and said, "Take the visitor to jail!"

Before Vivian could reply, the two burly lizard guards grabbed her arms and took her away. The guards led her down two flights of stairs to the basement of the building. The jail was located at the end of a dimly lit hall behind a purple steel door. The guards tossed Vivian into a dark, cramped cell that was built for lizard creatures, not humans. As she lay in the dark, Vivian thought that maybe babysitting and visiting relatives wasn't such a bad way to spend the summer after all.



ESCAPE

When her eyes got used to the light, Vivian saw that the only furniture in her cell was a small cot. There were two other lizard creatures in the cell next to hers. One was thin and the other was plump. They seemed to be arguing. As Vivian moved toward them, she discovered that they were playing a game. She heard this conversation:

Thin Lizard: Ready—one, two, three...

Both Lizards (together): AIR!

Plump Lizard (angrily): 3,974,425 to 72

Thin Lizard: Ready—one, two, three...

Both Lizards: LAND!

Plump Lizard: 3,974,426 to 72

Thin Lizard: Ready—one, two, three...

Both Lizards: AIR!

Plump Lizard: 3,974,427 to 72

Thin Lizard: Ready—one, two, three...

Both Lizards: LAND!

Plump Lizard: I quit! No matter what strategy I use, you always know what I'm going to say.

Thin Lizard: It's pointless to quit, Xax, since there's nothing else to do. Try a different strategy.



Plump Lizard: All right. But if you keep winning I'll quit even if it means we have to stare at the wall all day.

Vivian interrupted the lizard creatures and asked them what game they were playing.

"It's a war game," said the thin lizard creature, whose name was Xon. "The defender tries to guess how the invader will attack. I'm the defender and Xax is the invader. At the count of three, we each can say 'AIR' or 'LAND.' If I say the same thing as Xax, then I have guessed his attack strategy and I win. If we say different things — that is, if one of us says 'AIR' and the other says 'LAND' — he has avoided my defense and he wins. Right now I'm ahead by a score of 3,974,428 to 72."

Vivian asked Xax what kind of strategy he used in deciding how to attack.

"Well, first I was attacking by AIR all the time," said Xax, "but Xon got wise to that, so I started attacking by LAND all the time. He caught on to that too, so I started alternating and saying AIR-LAND-AIR-LAND but he figured that out as well. Now I don't know what to do."

"You could use a random strategy," Vivian told Xax. "That way it would be impossible to predict for sure whether you would invade by air or land. In the long run, Xon could only be right about half the time." Vivian then used her computer to demonstrate a random strategy.

We will write a program for the AIR-LAND invasion game in which the computer is the invader and you are the defender. Remember the rules: If you correctly guess the invader's route (AIR or LAND), you win. If your guess is incorrect, you lose. The computer will pick an invasion route at random so you will not be able to predict whether it will choose AIR or LAND.

The BASIC function RND(1) generates a "random" decimal fraction. A sequence of numbers generated this way has no predictable pattern. To see a sequence of these random numbers, enter and RUN this program. (If the number is very small, the computer will use exponential notation.)

```
10 PRINT RND(1)
20 FOR J = 1 TO 100: NEXT J
30 GOTO 10
```

You can eliminate the fractional part of a number by using the INT function. INT(X) drops the fractional part of X. For example, INT(3.1416) = 3 and INT(.3679) = 0.

If you multiply RND(1) by 2 you will get a random number between 0 and 2. You can change the random number 2*RND(1) into a "random digit," 0 or 1, by using the INT function to drop its fractional part. There is a

50-50 chance that this random digit will be 0 and a 50-50 chance that it will be 1. This is just like tossing a coin, in which there is a 50-50 chance that heads will come up and a 50-50 chance that tails will come up. To see some random digits, change line 10 as follows and RUN the revised program. The semicolon is included at the end of the PRINT statement so that the random digits will be printed across the line instead of in a column.

```
10 PRINT INT (2*RND (1) ) ;
```

We will instruct the computer to invade by AIR if the random digit $\text{INT}(2 * \text{RND}(1))$ equals 0 and to invade by LAND if $\text{INT}(2 * \text{RND}(1))$ equals 1. This is like tossing a coin and picking AIR if heads comes up and LAND if tails comes up. To do this, we will use the IF-THEN statement, which instructs the computer to follow an instruction if a specified condition is met.

In order to have a variable equal a specified value, we can use the LET statement. For example, if we wanted to have $X = 20$ we could use the instruction

```
LET X = 20
```

In fact, it is not even necessary to use the word LET. We could simply type

X = 20

and the computer would understand. In the next program we will let X equal a random number.

Clear the computer's memory, then enter and RUN the following program, in which the computer will print AIR if INT(2*RND(1)) equals 0 and will print LAND if INT(2*RND(1)) equals 1.

```
10 X = INT (2*RND (1) )
20 IF X = 0 THEN PRINT "AIR"
30 IF X = 1 THEN PRINT "LAND"
40 FOR J = 1 TO 200: NEXT J
60 GOTO 10
```

The computer now picks an invasion route at random. It can't tell you this in advance or you will know how to defend yourself, so it will keep the information secret, stored in its memory as the variable B\$. Enter these lines:

```
10 PRINT "AIR-LAND INVASION"
20 PRINT
25 X = INT (2*RND (1) )
30 IF X = 0 THEN B$ = "AIR"
40 IF X = 1 THEN B$ = "LAND"
```

After the computer chooses an invasion route, you must choose a defense. Add the following lines to the program. Line 65 ensures that you correctly enter AIR or LAND as your defense before the computer proceeds.

```
50 PRINT "INPUT DEFENSE"
55 PRINT " ('AIR' OR 'LAND' )"
60 INPUT A$
65 IF A$ <> "AIR" THEN IF A$ <> "LAND" THEN GOTO 50
70 PRINT
```

After this, the computer must tell you its invasion route and who won. The inequality symbols "<>" together, which are obtained by using the symbols "<" and ">", mean "not equal to." Add these lines and play the game:

```
80 PRINT "I CHOOSE TO INVADE"
85 PRINT "BY "; B$
90 PRINT
100 IF B$ = A$ THEN PRINT "YOU WIN"
110 IF B$ <> A$ THEN PRINT "I WIN!!"
120 PRINT
130 GOTO 25
```

When you are tired of playing the game, you may find it difficult to exit the program because of line 65. Holding down the **RUN/STOP** key and pressing the **RESTORE** key restores the computer to the state it was in when you first turned it on, except that the program is still available. (LIST it and see.)

Vivian suggested to Xax that on each play he could toss a coin to decide whether to attack by AIR or LAND. Xax thanked Vivian for her help. Then, he looked at her curiously and said, "By the way, who are you? You don't look like a lizard."

Vivian told Xax and Xon who she was and why she came to Planet X. When Xon heard about the help message he said, "You're the visitor from the sky! You're the one who Xavier promised would be sent to save us!"

Vivian asked Xon who Xavier was and what she was going to save them from. Xon replied, "Our civilization was once a prosperous one. Machines did our work for us. Propeller cars carried us wherever we wanted to go and farm machines harvested our crops. We had machines that cooked and machines

to build things. Our lives depended on machines. Then, we ran out of fuel. The Leadership Council didn't tell anyone about the shortage until it was too late. They hoarded what remained for themselves. They thought they would eventually find more fuel but they didn't."

"Now animals pull our cars and the machines are useless. The Leadership Council puts anyone who disagrees with them in jail. I'm here because I told people we should be looking for new kinds of fuel. Xax is in jail because he refused to work. He used to be a mechanic and now there are no jobs for mechanics. We belong to the Rebel Forces. Xaviar is our leader."

"We live in the mountains in caves. Xaviar knows magic and all of his wishes come true. He has told us that a visitor will come from the sky to save us. You are the visitor. Long live Xaviar! Long live Xaviar!"

Vivian was not sure how she was supposed to help Xaviar and his followers or, for that matter, anyone else on Planet X. She decided she had better try to escape from jail and get back to her spaceship. She went to the cell door and examined the lock. It was a combination lock marked with the Planet X numbers from 1 to 30. It worked like an Earth combination lock and would open after turning the lock to three numbers. Since there were 30 possible numbers for each rotation, there were $30 \times 30 \times 30$ or 27,000 possible number combinations, only one of which was correct. If it took 20 seconds to try a combina-

tion, it would take 150 hours or about one month working five hours a day to try all 27,000 combinations. Vivian knew that she couldn't spend that much time trying to open the lock. Fortunately, her pocket computer had a vibration detector so that it could detect when a correct number put the lock mechanism in alignment. Vivian attached the computer to the lock and within minutes she found the right combination. With a big smile, she opened the cell door.

We will write a program, using FOR/NEXT loops, that demonstrates the speed of the computer in trying all the 27,000 possible combinations. To start, we will make the second and third numbers we try equal to 1 and let the first number range from 1 to 30. Enter and RUN this program:

```
10 FOR J = 1 TO 30
20 PRINT J, 1, 1
30 NEXT J
```

Now we will let the second number range from 1 to 30 as well. Make the following changes and RUN this program:

```
15 FOR K = 1 TO 30
```



```
20 PRINT J, K, 1  
25 NEXT K
```

Finally, we will let the third number range from 1 to 30. Make the following changes and RUN the finished program.

```
17 FOR L = 1 TO 30  
20 PRINT J, K, L  
23 NEXT L
```

If you press the **RUN/STOP** key to stop a program, you can restart it by typing **CONT** and pressing **RETURN**. Try starting and stopping the lock combination program as it is being run.

When Vivian opened the lock to her cell, Xax called out, "Hey, what about us? Open our cell too." Vivian wanted to get back to her spaceship as quickly as possible, but she couldn't let Xax and Xon stay in jail. She attached her computer to the lock on Xax and Xon's cell and soon they were free. The three tiptoed down the dark hall and, when they were sure no one was looking, they dashed up the stairs and out a back door.

When she got outside the building, Vivian ran down the street with Xax and Xon close behind. As they rounded the corner, a group of lizard guards came out of the building.

"STOP!" shouted a guard as he and the other guards ran toward the street.

"WAIT!" shouted Xax and Xon as they ran after Vivian.

Vivian ran through a narrow alley and came out on a street lined with a series of small shops. She ran into a shop with a sign on the front that said "XELDA'S TOURS." Moments later Xax and Xon ran by the shop, not noticing that Vivian had gone inside. Soon after, the guards went running by.

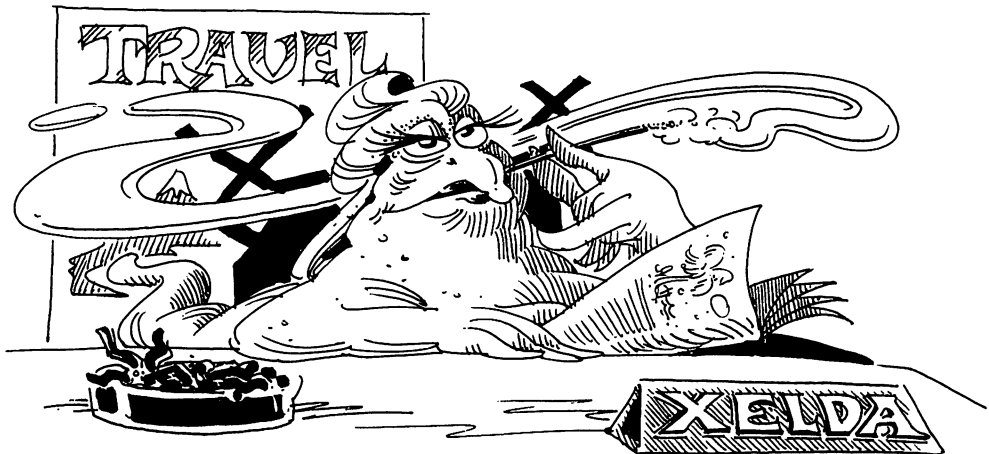
Vivian found herself in a room in which the walls were filled with pictures and maps. There was a picture of a lizard woman lying on a rock next to a swimming pool. Another picture showed a lizard family standing in front of a large hotel. There was another picture of what looked like a lizard amusement park.

At the far end of the room was a large table covered with more maps and pictures. Behind the table was a desk. Sitting at the desk was a creature who looked like an overweight hen. She was smoking a long green cigarette. On her desk was an ashtray filled with green cigarette butts. Her face was covered with multicolored makeup and she was wearing purple beakstick. The hen creature was reading a magazine called "Travel." There was a nameplate on her desk that said "XELDA."

"What can I do for you, honey?" said Xelda when she saw Vivian.

"Hi," said Vivian, looking nervously at the front door. "I'm wondering if you have tours to uhh... well, actually, I left my spaceship not far from here and I need a ride. And, if you can't do that, do you have any tours to Earth? You see, I need to get home in a hurry."

"I don't know where Earth is, but the xamster express leaves for the other side of the mountain every Monday morning," said Xelda as she put out her cigarette in the ashtray and lit another one. "We also have a scenic 10-day cruise for 150 xents."



Get away from the pressures of the city. There's dancing on deck every night to the live music of Xandy and the Leaping Lizards. Every day there's a new activity. You can meet new friends or bring your own. I'd go myself if I could find someone to run the store. Due to the fuel shortage, we have reduced rates on every tour. Here's a pricellst." Xelda handed Vivian a pricellst.

"It's hard to make a living in this business without any fuel," she said. "I should have stayed in souvenirs." Xelda put out her long cigarette and lit another one.

Vivian looked at Xelda's pricellst. It had ten tours listed as follows (The Planet X symbol for xents is "x"):

1. Around the mountain 75x
2. 10-day cruise 150x
3. Scenic walk 40x
4. Xamster express 50x
5. Lake Lizard tour 60x
6. Cave exploration 35x
7. Week under the suns 80x
8. Town tour 100x
9. Fun on the rocks 60x
10. Creature comforts 40x

"Very interesting," said Vivian. "I wonder if you could tell me..." Vivian didn't have time to finish her sentence, however, because just then Xax and Xon came running through the door.

"There you are!" said Xax as he grabbed Vivian by the arm.

"Sorry for the disturbance," said Xon to Xelda, as he grabbed Vivian by the other arm.

"Stop!" said Vivian as she struggled to get free. "I want to go back to my spaceship."

"You can't leave now," said Xon. "You're the visitor from the sky who will help us. You must come with us to see Xavlar."

"I've got to move to a better neighborhood," said Xelda as she resumed reading her magazine.

Xax and Xon took Vivian outside to a parked propeller car. "This one has fuel in it," said Xax as he picked Vivian up and put her in the front seat. "Let's get out of here before the guards come."

Computers are often used to keep track of things like pricelists. We will write a program that accesses Xelda's tour pricelist.

An ordered list of numbers is called an array. To store an array in the computer's memory, you must first

dimension it so that the computer will know how many “elements” or numbers there are in the list. The *dimension (DIM) statement* is used to indicate the maximum number of elements in the array. The computer starts counting elements with element #0. (Actually, in Commodore 64 BASIC, if you will have 11 or fewer elements in the array, you don’t need a DIM statement, but it is good practice to always use it.) For example, if there will be no more than 30 elements in an array, you can use a dimension of 29 (or any dimension larger than 29). An array name is like a name for a numeric variable. It must begin with a letter and can be followed by a letter or a number. To specify a certain element in the array, you give the array name followed by the number of elements in parentheses. For example, if the tour pricelist array is denoted by P, the price of the sixth tour would be P(6).

The following program will store Xelda’s tour pricelist and enable you to find the price for one or more creatures to go on a particular tour. The array begins with P(0), but we will use P(1).

```
10 DIM P(10)
20 P(1) = 75:P(2) = 150:P(3) = 40
30 P(4) = 50:P(5) = 60:P(6) = 35
40 P(7) = 80:P(8) = 100:P(9) = 60
```

```
50 P(10) = 40
55 PRINT
60 PRINT "WHICH TOUR # DO YOU WANT? "
70 INPUT T
80 IF T > 10 THEN GOTO 60
90 IF T < 1 THEN GOTO 60
100 PRINT
110 PRINT "HOW MANY WILL BE GOING? "
120 INPUT N
130 PRINT
140 PRINT "THE PRICE OF TOUR "; T;
150 PRINT " IS "; P(T) ; " XENTS"
160 PRINT
170 PRINT "YOUR TOTAL COST WILL BE ";
180 PRINT N*P(T) ; " XENTS"
```

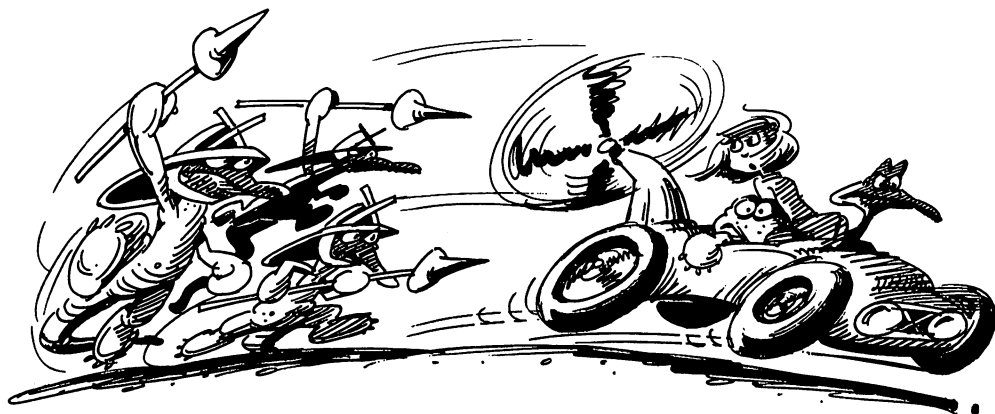
You can easily change the values of individual elements in arrays. You can also expand an array into a matrix in which there is more than one number for each item in the list. For example, Xelda's pricelist could have a regular price and a discounted price for each tour. To have two prices for each tour, you would have a list of 10 pairs of prices. This information would be stored as a "10 by 2 matrix" and would be given the dimension DIM P(10,2). For example, the regular price and discount price of tour 7 would be P(7,1) and P(7,2) respectively.

FLIGHT TO THE HILLS

Xax tried to start the propeller car but the engine only sputtered and stopped. Just then, three lizard guards came running around the corner. "Hurry!" Xon urged Xax.

Xax tried to start the car again. After more sputtering, it started.

We will write a program in which the computer makes the sound of the propeller car. In Commodore 64 BASIC,



sound is created using certain POKE statements.

A POKE statement such as POKE 54296,15 finds the location in the memory of the Commodore 64 whose “address” is 54296 and “pokes” the value 15 into it. When you first turn on the computer, the value stored in address 54296 is 0.

Now type:

POKE 54296, 15

and hit **RETURN** . The number 15 is now stored at address 54296. You will hear a little “click” after you hit **RETURN** . This is the sound of your TV speaker being “tugged,” that is, pulled by the Commodore sound system.

How a computer behaves depends on what numbers are “stored” in the various memory locations. Address 54296 accepts any number from 0 to 15 and determines how loud the music and sound effects will be (0 is inaudible, 15 is loudest). Addresses 54276, 54283, and 54290 determine what kind of sound (or “waveform”) is played by these three “voices.” A value of 129 creates “white noise” for sound effects, while 128 shuts it off.

When you first turn on the Commodore 64, all these locations have 0 stored in them.

Addresses 54272, 54273, and 54278 also contain information for the voice at 54276. We will explain these later. For now, type and RUN this program:

```
5 FOR Z = 1 TO 24
10 POKE 54296, 15
20 POKE 54278, 248
30 POKE 54273, 17: POKE 54272, 37
40 POKE 54276, 129
50 FOR T = 1 TO 100: NEXT T
60 POKE 54276, 128
70 NEXT Z
```

By the way, you may STOP the program and be stuck with a tone. To eliminate it without turning off the computer, just type

```
POKE 54296, 0
```

(and hit **RETURN** , of course), or hold down **RUN/STOP** and **RETURN** , which restores the “initial” values into the memory.

To let the engine run indefinitely, add the following:

```
80 GOTO 5
```

To stop the engine, hold down **RUN/STOP** and press **RETURN** .

Next, you can change the “speed” of the engine by varying the length of delay in line 60. Make these changes and input different numbers for the the delay (Use a delay of 30 to start with).

```
3 INPUT B
50 FOR T = 1 TO B: NEXT T
```

The lizard guards ran across the street just as the propeller car lurched forward and around the corner. The car almost crashed into a listless crowd of lizard creatures, narrowly missed a giant xamster, but managed to take Vivian, Xax, and Xon to the edge of town. Soon they were bouncing along a bumpy dirt road on their way into the mountains.

Commodore 64 BASIC allows you to create graphics on the screen. We will illustrate the route of the propeller car. The TV screen is divided into 25 rows and 40 columns.

Clear the computer's memory by typing NEW. Then enter and RUN the following two programs:

```
10 FOR N = 1 TO 25
20 PRINT N;
30 IF N = 25 THEN GOTO 30
40 PRINT
50 NEXT N
```

You will have to use the **RUN/STOP** key to keep line 30 from returning to itself. The purpose of that line is to keep the computer from printing "READY." Now type NEW and enter and RUN this program:

```
10 FOR N = 1 TO 40
20 PRINT "A";
30 NEXT N
```

There are 40 A's across the screen. You can see this in one of two ways: either count them or look at line 10 of the program.

You can print graphics characters on the screen in the graphics mode by using the **SHIFT** key (which gives you the characters on the right) or the **⌘** key (which gives you the characters on the left) while press-

ing any key with graphics characters on the front. Try this with the various keys.

We will use the character  to represent the propeller car. You get this character by holding down the  key and pressing the + key.

You can move the cursor around by using the  and  keys. The lower direction works if you press the key; the upper one works if you use the  key at the same time. We will use the notation CRSR↑, CRSR↓, CRSR←, and CRSR→ to denote the different directions of the cursor keys.

Commodore 64 BASIC allows you to put these cursor movements into the program. As an example, press the  key along with the  key. That clears the screen. Now clear the computer's memory and enter the following program:

```
10 PRINT "  
```

By this we mean: strike the  key with the  key pressed, after the first quotation marks, and then close the quotes. RUN this program. It clears the screen just as the direct command would have.

The symbol (reverse color heart) that appears inside the quotes is just an abbreviation for **SHIFT CLR/HOME**. All the direct commands (color change, cursor movement, etc.) have similar abbreviations.

Here is a short program using cursor movements:

```
10 PRINT "CRSR ↑ [3 SPACES] I RAN AL- "
```

Clear the screen and RUN the program.

The [3 SPACES] means to leave three spaces at that place in the program.

Line 10 moves the cursor up to the line where you typed RUN, erases RUN by typing three blank spaces, and then puts "I RAN AL-" on the screen which connects to the READY two lines down, which the computer types automatically when a program finishes. To make the display even neater, we could move the cursor back before typing "I RAN AL-". This can be done as follows:

```
10 PRINT "CRSR ↑ [3 SPACES] CRSR ← [3X] I RAN AL- "
```

Try it.

[3X] means to type CRSR ← 3 times.

We will use this technique to plot the path of the propeller car. We will have the car flash on and off and "move" across the screen. This kind of motion is called

animation. Cartoons are made in much the same way. In animation, nothing actually moves. Instead, a series of pictures is drawn and erased very fast. This gives the appearance of movement. To erase a picture, we will move the cursor back one space and draw a blank space. Type NEW to clear the computer's memory. Then enter and RUN this program, typing + while the  key is pressed.

```
30 PRINT "  + ";
```

Now we will have the car flash on and off by erasing and redrawing it at the same location. Make these additions and RUN the revised program. The delay statement makes the flashing effect clearer. The underline symbol in line 50 indicates that one space should be left.

```
20 FOR N = 1 TO 39
40 FOR T = 1 TO 100: NEXT T
50 PRINT "CRSR ←_ CRSR ← ";
55 FOR T = 1 TO 100: NEXT T
60 NEXT N
```

Line 50 moves the cursor back, prints a blank space, and then moves the cursor back, ready for line 30 to print it again.

Now we will have the propeller car move across the screen. We change line 50 so that it doesn't back up after erasing. Typing line 55 and pressing **RETURN** erases line 55.

```
50 PRINT "CRSR←_ ";  
55  
70 PRINT "  + ";
```

RUN the program.

Now we will make the propeller car go across the screen and return. To do this the computer must perform the above sequence of operations and then do it in reverse. Add the following lines and RUN the program. In line 80, [2X] means to type the previous symbol, CRSR ←, two times.

```
20 FOR N = 1 TO 40  
70 FOR N = 1 TO 39  
80 PRINT "CRSR←_ CRSR←[2X]  + ";  
90 FOR T = 1 TO 100: NEXT T  
100 NEXT N
```

To add the sound of the propeller car we will use what is called a subroutine. A subroutine may be useful when you have a set of instructions that you want

to access more than once in a program. To access a subroutine you use a *GOSUB* statement, which directs you to the line number that begins the subroutine. At the end of the subroutine you use a *RETURN* statement, which sends the computer back to the line after the *GOSUB* statement. Using *GOTO* statements will accomplish the same thing but, if the subroutine is accessed many times in a program, the *GOSUB* and *RETURN* statements are more convenient.

Add the following lines to the program. List the program so you can see the way the subroutine is accessed by the *GOSUB* statements then *RUN* the program. The *END* statement in line 110 tells the computer to stop before it runs into the subroutine after it is finished with the rest of the program.

```
10 POKE 54296, 15
12 POKE 54278, 243
14 POKE 54273, 17: POKE 54272, 37
40 GOSUB 500
90 GOSUB 500
110 END
500 POKE 54276, 129
510 FOR T = 1 TO 100: NEXT T
```

```
520 POKE 54276,128
530 RETURN
```

Finally, we will make the propeller car travel back and forth and move down the screen. To do this we use a few cursor movements that cause the propeller car to be erased and reprinted on the next line each time it reaches the left edge of the screen. Make the following revisions and RUN the revised program:

```
110 PRINT "CRSR ← _ CRSR←CRSR ↓ ";
120 GOTO 20
```

To stop the program hold down **RUN/STOP** and press **RESTORE** .

Riding along the dirt road leading into the hills, Vivian noticed changes in the landscape. There were piles of large, jagged rocks along the roadside that looked like they might tumble down at any moment. Tall, purple trees partially blocked the light coming from the sun. The climb grew steeper. Every so often, Vivian heard animal noises coming from behind bushes or rocks. The road got narrower until, finally, the rocks

and trees prevented the propeller car from going any further. Xax stopped the car.

"Are you sure this is the right road?" Vivian asked Xax.

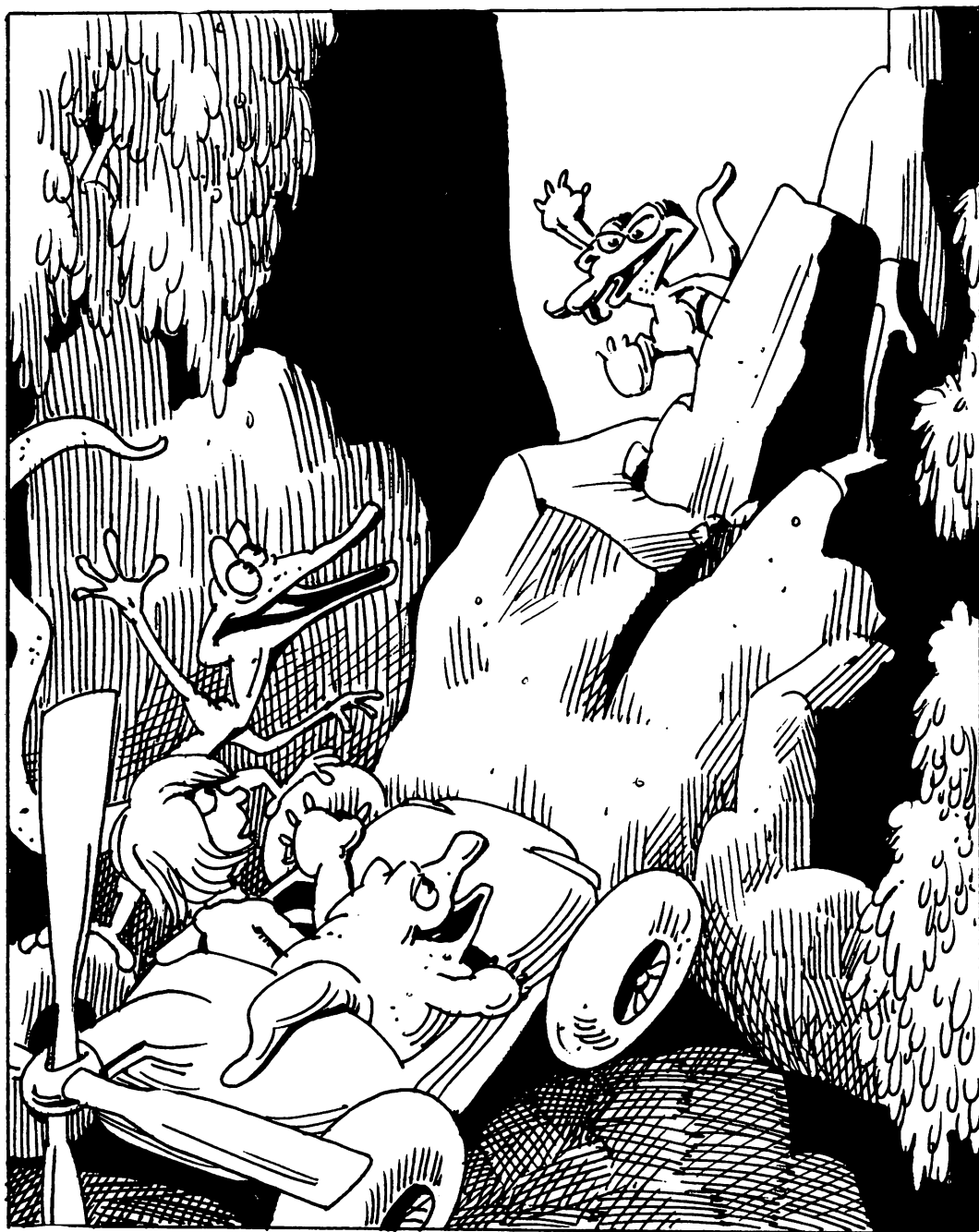
"I think so," he said. Xax made a high whistling sound. A few moments later, a thin lizard with slick hair and a mustache darted out from behind a pile of rocks.

"Xax! Xon!" said the thin lizard creature. "Welcome home!" The lizard creature then looked at Vivian uncertainly.

"Xete, I'm so glad to see you!" said Xon as he embraced the thin lizard creature. Xon motioned toward Vivian. "This is Vivian," said Xon. "She is the visitor from the skies who has come to save Planet XI"

"I hope you're right," said Xete suspiciously. "We must take her to Xavlar without delay!"

Before Vivian could reply, Xete took her hand and led her through a thicket of bushes with Xax and Xon close behind. Soon Vivian and her companions were walking along a narrow path through a thick forest of purple trees.



XAVIAR

The small path Vivian and the lizard creatures were on became even narrower. Vivian was becoming exhausted. Branches with thorns brushed across her face and she had to stoop to avoid overhanging branches. Purple plants left a sticky substance on her clothes. She shivered with cold. It was late in the day and the sun was getting lower in the sky.

They were high in the hills now. Vivian wanted to stop and rest. She walked around a turn and the path came to an end, blocked by thick bushes.

"We're here at last," said Xon. Pulling apart some branches, he stepped into the bushes and disappeared from sight. Xete followed Xon through the opening and Xax motioned for Vivian to follow Xete. With difficulty, she squeezed her way through the hole in the bushes.

On the other side of the bushes was a clearing. At the far side of the clearing a jagged mountain of rocks extended out of sight into a purple mist. A large campfire was in the middle of the clearing. Busy lizard creatures hurried back and forth from the clearing to the caves at the base of the mountain. As Vivian walked toward the campfire, the lizard creatures came to greet her companions.

When they reached the campfire, a large, handsome lizard with thick, wavy hair walked over and said, "Welcome back Xax and Xon, I see that you brought a visitor."

"Yes we have," said Xon. "Xaviar, this is Vivian, the visitor from the skies whom you promised would come."



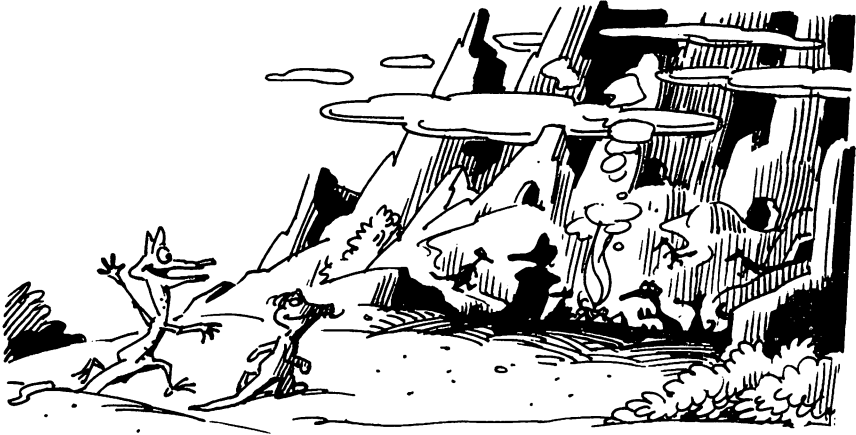
"I'm so glad you're here," said Xaviar. "I knew you would come because I wished that you would, and all my wishes come true. You are here to find fuel for us so that I can become the ruler of Planet X. By the way, what took you so long?"

Vivian told Xaviar that the only reason she came to Planet X was because she received his help message, but that after giving the matter some thought, she really didn't think she could help him.

"Help message?" said Xaviar. "I didn't send any help message ... But anyway, who cares? Now that you're here, you will find fuel for us." Xaviar motioned with his hand. "See the beautiful landscape. See the Monday sun setting in the sky. We all want to live happily here. We need rulers like me who understand the needs of the citizens. We need fuel."

We will write a program that displays the Planet X landscape and the changing colors of the sky caused by the setting Monday sun. To do this, we will use the POKE command. In Commodore 64 BASIC, we can change certain aspects of the screen and border colors just by poking numbers into the right addresses.

We will create the Planet X landscape by using a graphics character that looks like a tree. To get the graph-



ics character on the right side of a key, hold down the **SHIFT** key and press the key you want. To get the character on the left side of the key you hold down the **⌘** key and press the key you want. (In Chapter 5, we used the **⌘** key to print the propeller car graphics.)

The keys that have graphics characters are the letter keys along with “+”, “-”, “£”, “@”, and “↑”. Try typing these characters. Notice that the right-hand character for the “A” key looks like a tree. (Actually, it is the “spades” symbol used in displaying cards.) We will use this symbol to create a landscape of trees.

The first thing we will have the program do is print the landscape. Enter and RUN this program:

```
10 FOR I = 1 TO 999
20 PRINT " SHIFT A";
30 NEXT I
40 GOTO 40
```

Line 40 keeps the computer from skipping a line and printing READY. We are missing one tree in the corner because, the way we are doing it, the computer would skip a line after printing the last tree.

Next we will use the POKE command to illustrate



the colors caused by the setting sun on Planet X. Address 53280 determines the border color and 53281 determines the background color of the screen. They accept numbers from 0 to 15. (For a table of the colors see the Glossary).

Add these lines and RUN the revised program:

```
40 FOR B = 0 TO 15
50 FOR C = 0 TO 15
60 POKE 53280, B
70 POKE 53281, C
80 FOR M = 1 TO 2000: NEXT M
90 NEXT C
100 NEXT B
110 GOTO 110
```

When you hit **RUN/STOP** the screen and border will probably not be “normal,” but at whatever setting it was last “poked” to. To restore the original hold down **RUN/STOP** and press **RESTORE**.

Finally, we will add sound to the program. Planet X is inhabited by a small insect called a “xricket” that makes a noise surprisingly like the chirp of a cricket. We will add a subroutine to the program that simulates the sound of a xricket. Instead of making the xricket

chirp at regular times we will access this subroutine at "random." We can do this with the RND(1) statement. We will pick a number from 0 to 2 at random and access the chirp subroutine only if the selected random number is 1. This means that there is a 1 in 3 chance of hearing the cricket chirp whenever line 52 is executed. Add the following lines, and RUN the final version of the program. Line 2 clears the sound addresses.

```
2 FOR L=54272 TO 54296: POKE L, 0: NEXT L
3 POKE 54296, 15
4 POKE 54277, 3
72 R = INT(3*RND(1))
75 IF R = 1 THEN GOSUB 200
200 FOR Z = 1 TO 5
210 FOR X = 220 TO 250 STEP 30
220 POKE 54273, X-80: POKE 54272, X-80
230 POKE 54276, 17
240 NEXT X
250 POKE 54276, 16
260 NEXT Z
270 RETURN
```

Line 210 is the STEP variation of the FOR/NEXT loop. In this case X counts by 30 and so assumes two values: 120 and 150. We will explain the other POKEs later.

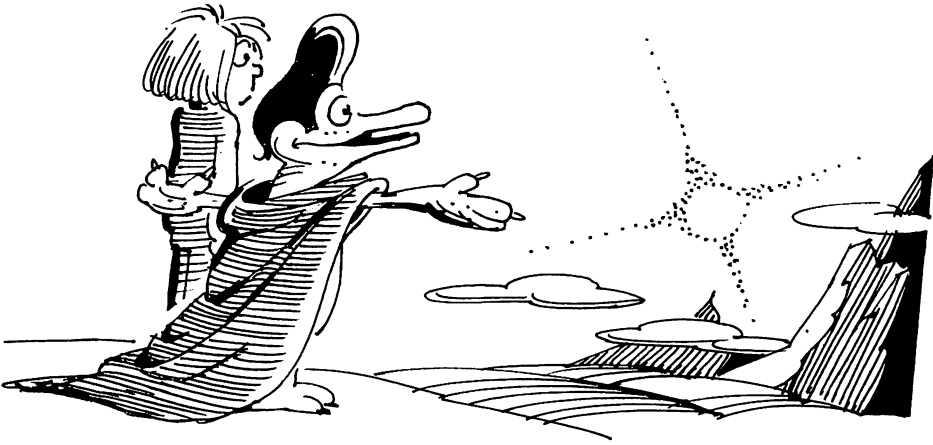
As the Monday sun set in the Planet X sky, Vivian turned to Xaviar and said, "It will be dark soon. Tomorrow I'd like to go to my spaceship and be on my way home. I have to get back to Earth before school starts."

"It will be dark soon?" said Xaviar. "I'm not sure what you mean. It is only dark in caves and when we put blankets over our heads."

"Doesn't it get dark after the sun sets?" Vivian asked Xaviar.

"Don't be silly," said Xaviar, who then told Vivian about time on Planet X. "You see," he said, "there are two suns in the Planet X system, Monday and Yadnom. When the Monday sun sets, the Yadnom sun rises so that one sun is always in the sky. As one sun goes across the sky, the other sun goes under the planet to get ready for the next day. Thus, there are only two days on Planet X, Monday and Yadnom, named after the suns. Today is Monday and tomorrow it will be Yadnom. After that it will be Monday again."

"We have two clocks," continued Xaviar, "one for each sun. The unit of time is the XSEC. There are 100 XSECs in a day. Each day goes from 100 to 200 o'clock. When a sun rises, the clock for that sun starts at 100 o'clock. When the sun sets it is 200 o'clock. When the sun has set, time for that sun goes backwards so that the next day will begin as usual at 100 o'clock." Xaviar looked at the sky and pulled a small watch from his pocket. "It is now 199 Monday time and 101 Yadnom time. In another XSEC, the Monday sun will set and the Yadnom sun will appear."



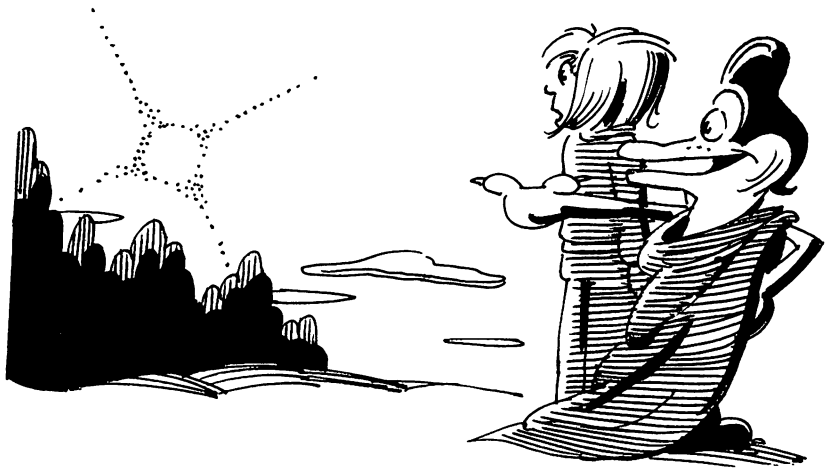
Moments later, the Monday sun began to disappear behind the horizon. As this happened another sun appeared on the other side of the sky. "There it is," said Xaviar, "the Yadnom sun." Xaviar looked at his watch. "It is now 200 Monday time and 100 Yadnom time. Good morning. Good evening. It is Yadnom morning. It is also Monday evening. Monday time goes backwards now and Yadnom time goes forward. It is now 199 Monday time and 101 Yadnom time. Soon it will be time for Monday dinner, which is also Yadnom breakfast. I hope you will join us."

We will write a program that simulates a Planet X digital clock. In order to quickly demonstrate the forward-backward feature of Planet X time, an XSEC in our computer clock can't be quite as long as a real XSEC. By experimenting with delay statements, you can create an actual Planet X digital clock or even an Earth clock.

In writing this program, we will use READ and DATA statements. These statements allow you to store numerical or written information in a program. Whenever a READ statement is encountered, the computer will read as many times as are specified from a DATA

statement located at the end of the program. The number of DATA items requested by the READ statement can't be more than the number of items listed in the DATA statement unless a RESTORE statement is also used. The RESTORE statement instructs the computer to go back to the beginning of the DATA list for more DATA to read. In our clock program, the computer will read through the list of data twice so we will need to use a RESTORE statement.

To practice using the READ, DATA, and RESTORE statements, enter and RUN the following program. The



program instructs the computer to read pairs of numbers and print them. The RESTORE statement causes the program to go back through the list of data when it is finished the first time.

```
10 FOR K = 1 TO 2
20 FOR I = 1 TO 2
30 READ A, B
40 PRINT A, B
50 NEXT I
60 RESTORE
70 NEXT K
80 DATA 6, 7, 8, 9
```

Clear the computer's memory and then RUN the following program, which will create a Planet X digital clock.

```
10 PRINT TAB(10); "PLANET X TIME CLOCK"
20 FOR N = 1 TO 18
30 PRINT
40 NEXT N
50 PRINT TAB(5); "MONDAY"; TAB(30); "YADNOM"
60 PRINT : PRINT
62 PRINT TAB(5); "XSECS"; TAB(30); "XSECS"
```

```
64 PRINT "CRSR ↑ [3X] ";
70 FOR Y = 1 TO 2
80 READ A, B, C
90 FOR N = A TO B STEP C
100 PRINT TAB(5); N; TAB(30); 300-N
110 PRINT "CRSR ↑ ";
140 FOR Z = 1 TO 99: NEXT Z
150 NEXT N
160 NEXT Y
170 DATA 100, 200, 1
180 DATA 200, 100, -1
190 RESTORE: GOTO 70
```

The TAB(X) function tells the computer where to start printing. The first space is TAB(0) and so on up to TAB(39). Note that line 180 makes 90 read "FOR N = 200 TO 100 STEP -1." This means that N = 200, then N = 199, then 198, and so on.

To put sound in the clock, add the following lines:

```
120 POKE 54296, 15
130 POKE 54273, 129: POKE 54272, 100
135 POKE 54276, 17
145 POKE 54276, 16
```

"So you see," Xaviar continued, pointing toward the Yadnom sun, "there's always a sun in the Planet X sky. On very rare occasions, the two suns cross in the sky at once. It is said that strange things happen then."

Just then, a lizard creature announced that Monday dinner-Yadnom breakfast was ready. Vivian sat down to eat with Xaviar and the other creatures, wondering why the lizards had time go backwards. The A.M. and P.M. system worked perfectly well on Earth, she thought. She bit into a small green fruit that tasted like a sour lime and realized that she was very tired and needed to get some sleep.

THE TROLLEY

Vivian finished her Monday dinner-Yadnom breakfast and thanked Xaviar for his hospitality.

"I'm glad you enjoyed your meal," said Xaviar. "There are advantages to living in the mountains, but it's very isolated. We seldom have visitors, especially from outer space."

"I'm tired," said Vivian with a yawn. "I'd like to take a nap. Tomorrow I've got a long way to travel, about 100,000,000 miles."

"Xete will show you an empty cave," said Xaviar.

Vivian thanked Xaviar and walked with Xete to the base of the mountain, where a network of caves provided homes for Xaviar and his followers. Xete showed the tired girl a small, cozy cave containing a bed made of purple moss. In a few XSECs, Vivian was fast asleep.

When Vivian woke up a sun shone brightly in the sky. Of course, one of the Planet X suns always shone brightly in the sky.

Outside the cave, Vivian found Xaviar talking to Xon and another lizard creature. "Good afternoon-good night," said Xaviar with a toothy lizard smile. "I hope you had a good sleep."

"I slept quite well, thank you," Vivian said. "But I have to be going soon. I was wondering if someone could show me how to get back to my spaceship."

"I'm afraid that's impossible," said Xaviar. "You see, I wished that you would come to Planet X and find fuel and, as you know, all my wishes come true."



"Why don't you just wish that you would find some fuel by yourself?" said Vivian. "I really have to get going."

"Don't get upset," said Xavier. "I can't just wish anything. It has to be the proper type of wish. Anyway, I'll see that you get back to your spaceship, but first you must find fuel." Xavier motioned to a husky lizard creature who walked over to Vivian and took her by the arm.

"Xavier is our leader," said the husky lizard creature. "We do as he wishes and his wishes always come true."

Vivian knew that she was in a difficult situation. She couldn't get back to her spaceship without Xavier's help and he wouldn't do this unless she helped him find fuel. She was upset that she still didn't know who sent the help message. She thought about her predicament. Then she had an idea.

"Well, Xavier," she said. "I've decided that I'll help you find fuel."

"Good!" exclaimed Xavier. "Great!" exclaimed all the lizard creatures standing near Xavier.

"But, in order to help you find fuel," Vivian continued, "I must use my spaceship's laser scanner. I'll have to fly low over the countryside and scan for underground fuel deposits."

"Very well," said Xavier, "but in order to make sure that you don't just leave, I'll go with you in your spaceship. Besides, I've always wished that I could ride in a spaceship. I may as well make that wish come true."

After conferring with the other lizard creatures, Xaviar turned to Vivian and said, "Xax, Xon, Xete, and Xerxes will go with us to your spaceship. Only you and I will ride in the spaceship. There used to be a trolley system that ran up and down the mountain. On the other side of the clearing is an abandoned trolley. We have a small amount of fuel saved for special occasions. We'll ride the trolley down the mountain. We must leave now to reach your spaceship by tonight-tomorrow morning. Follow me." With Xax, Xon, Xete, Xerxes, and Vivian running to keep up with him, Xaviar walked quickly across the clearing to a grove of tall trees.

The abandoned trolley was on the other side of the grove. It was sitting on a rusty track that went out of sight into the bushes.

"Here it is," said Xaviar. He walked into the bushes and emerged a few moments later carrying two large cans filled with fuel. "It's down hill most of the way," said Xaviar as he poured fuel into the trolley's fuel tank.

"All aboard," said the lizard leader when he was finished, ringing a rusty bell. Vivian and the rest of the lizard creatures climbed aboard the trolley. After a few unsuccessful attempts, Xax started the trolley's engine and with a sudden lurch forward, the trolley slowly chugged off into the bushes.

The trolley bounced through the forest along the rusty track, showering its riders with leaves and broken branches. The lizards cheered excitedly while Vivian crouched behind a seat

to avoid the flying foliage. They sped precariously down the mountain. The trolley came out of the woods and across a large field. At the other side of the field Vivian had a spectacular view. Below them were hills and jagged rock formations. Farther down, Vivian could see a large lake and below that she could see the city. Vivian would have some interesting stories to tell her friends on Earth, providing that she was able to get away from this strange planet.

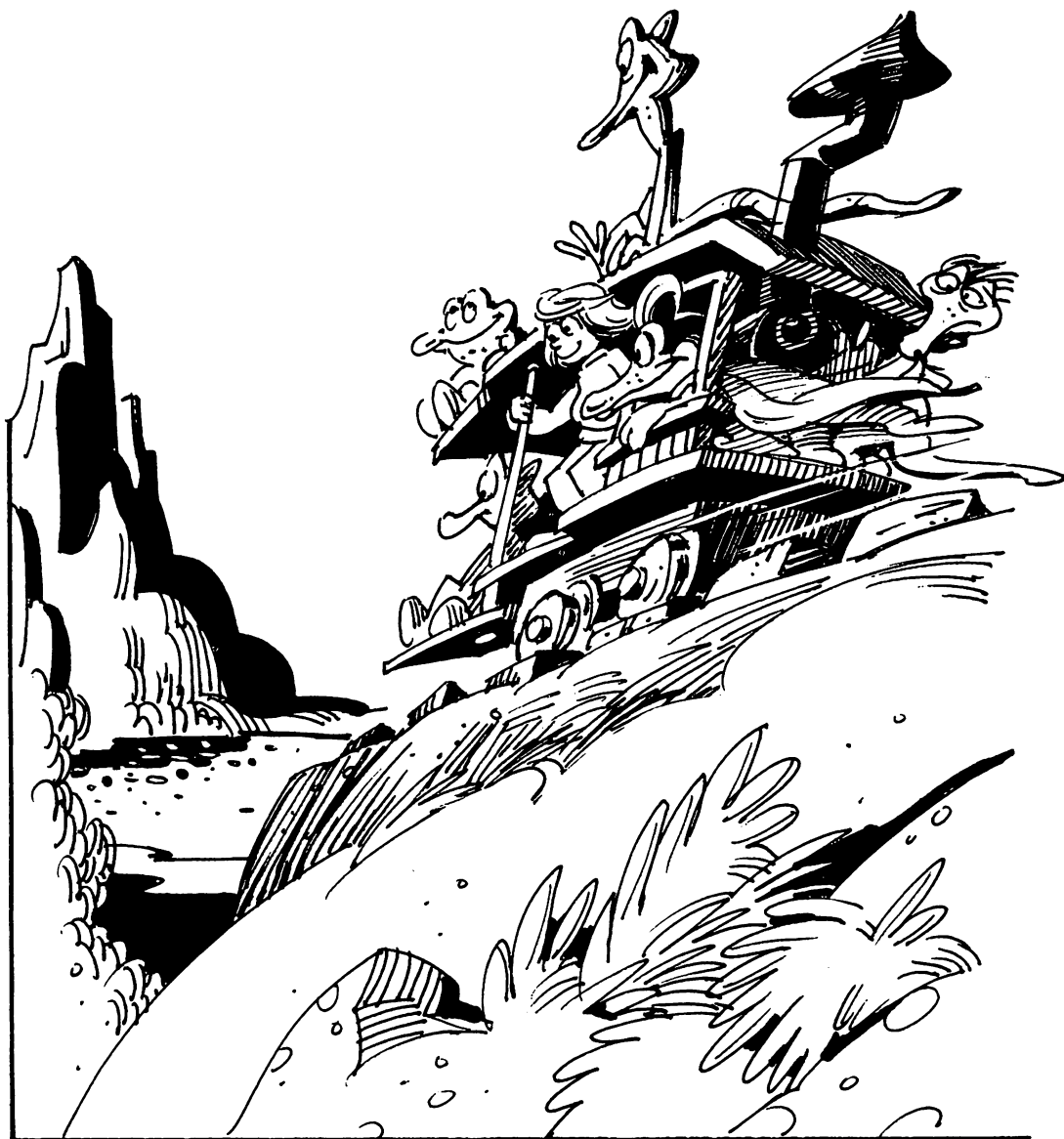
Traveling down a hill, Vivian watched the beautiful landscape pass by. Tall trees adorned the hills. The lake grew closer and it seemed to Vivian that if the trolley continued on its present course it would go straight into the water.

"How are we going to cross the lake?" Vivian asked Xavier. "Is there a boat somewhere?"

"Don't worry," said Xavier. "There's a bridge around the next bend."

Sure enough, as the trolley careened around the next bend, the tracks led over the lake along a narrow bridge. The bridge hadn't been traveled on in quite some time and, as the trolley carried them over the lake, the bridge started creaking and swaying. Vivian nervously looked at the water below. She glanced at the lizard creatures and saw that they were also looking at the water. She looked straight ahead and was horrified to see that in the middle of the lake the bridge had caved in.

"Stop the trolley!" Vivian shouted. "The bridge has collapsed!"





We will write a program that illustrates the predicament of the passengers on the Planet X trolley. First we will draw the track, using the graphic character "-" which we get by pressing **⌘ Y**.

```
10 PRINT " ⌘ CLR/HOME "; : PRINT "CRSR↓[6X] "  
30 FOR N = 0 TO 30: PRINT " ⌘ Y"; : NEXT N  
200 GOTO 200
```

Now we draw the trolley. Since we will draw this figure many times, we will put it in a subroutine:

```
60 PRINT: PRINT "CRSR↑CRSR↑";  
70 GOSUB 500  
500 PRINT " ⌘ Q [3 spaces] ⌘ Q CRSR←[5X] CRSR↑";  
510 PRINT " ⌘ + [5X] CRSR←[5X] CRSR↑";  
520 PRINT " ⌘ + _⌘ + _⌘ + CRSR←[5X] CRSR↑";  
530 PRINT " ⌘ + [5X] CRSR←[5X] ";  
540 RETURN
```

Now we will animate the trolley by erasing the left side (subroutine 600) and redrawing the trolley one space over:

```
80 FOR A = 0 TO 32  
90 GOSUB 600: GOSUB 500: NEXT A
```

```

600 FOR Z = 1 TO 4
610 PRINT "_ CRSR←CRSR↓";
620 NEXT Z
625 PRINT "CRSR↑_ ";
630 RETURN

```

To make the trolley fall, we precede 500 with a line that erases the top line of the trolley and positions the cursor to draw the trolley one line lower:

```

100 FOR B = 1 TO 18: GOSUB 490: NEXT B
490 PRINT " [5 spaces] CRSR←[5X] CRSR↓[4X] ";

```

Finally, we sink the trolley. Add the following lines and try to follow the logic before you RUN the program.

```

110 PRINT " [5 spaces] CRSR←[5X] CRSR↓[3X] ";
115 GOSUB 510
120 PRINT " [5 spaces] CRSR←[5X] CRSR↓[2X] ";
125 GOSUB 520
130 PRINT " [5 spaces] CRSR←[5X] CRSR↓ ";
135 GOSUB 530
140 PRINT " [5 spaces] ";

```

To add sound to the program, enter the following lines:

```
20 POKE 54296, 15
22 POKE 54278, 243
24 POKE 54277, 159
26 POKE 54273, 17: POKE 54272, 37
82 POKE 54276, 129
84 FOR X = 1 TO 100: NEXT X
86 POKE 54276, 128
```

The lizard creatures were busy watching the water and didn't notice that the bridge had collapsed until it was too late. Xax tried to apply the brakes but the trolley went off the end of the bridge and fell into the water below. Fortunately, the water was shallow. In fact, when a soaking wet and very upset Vivian climbed out of the trolley, she discovered that the water barely reached her knees.

A few XSECs later, Vivian and the lizard creatures were standing on the bank of the lake staring gloomily at the partially submerged trolley.

"I guess we should start walking," said Xaxiar. "When we get to the edge of town we will get a propeller car and drive to the spaceship." After a brief rest, Vivian and the lizards began hiking toward the city.



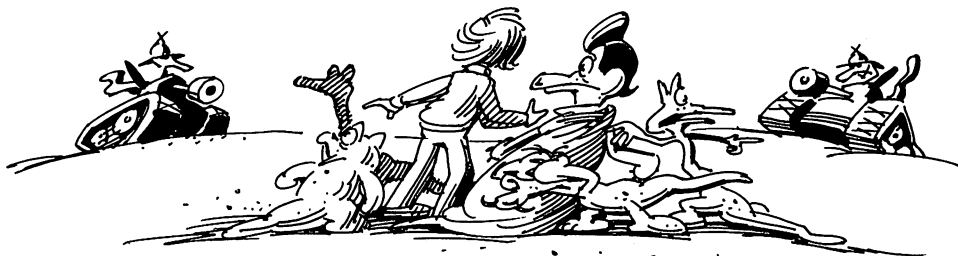
THE ARMY LIZARDS

Wet, tired, worried, sore, and angry, Vivian trekked through the woods with her lizard companions. The Yadnom sun sank lower in the sky and Vivian knew that it soon would be Monday again. She was trying to think of a way to get into her spaceship and take off without Xaviar when suddenly Xerxes turned and shouted, "Hide! Soldiers!"

The lizards dropped to the ground and crawled quickly into the bushes and under rocks. Vivian dropped to the ground also, but she could not slither the way lizard creatures could. Instead, she just lay still.

Soon Vivian heard a rustling sound in the bushes near her. She looked up and saw a group of lizard soldiers march by. She put her head down and held her breath until they passed. When she was sure the soldiers had gone, Vivian stood up and her companions came out from their hiding places.





"Be careful," whispered Xaviar as he motioned for the others to follow. Vivian crouched down as she followed Xaviar through the woods.

The girl and the lizards crept through the woods for what seemed like a very long time until they finally reached a clearing.

"On the other side of the clearing there is a trail that leads down to a farm," said Xaviar. "The farmer who lives there is an old friend of mine. We can use his propeller car. If he has no fuel we can hitch the car to a xamster. From there we'll go to the spaceship. Let's cross the clearing quickly."

Halfway across the clearing, Vivian heard a loud rumbling sound. Suddenly, a large tank appeared at one end of the clearing. The lizards in the tank saw Vivian and her companions and started to move toward them.

"Run!" shouted Xaviar above the roar of the tank. Vivian and the lizards started to run and the tank started shooting. With an enormous BOOM! a shell exploded near Vivian. Another explosion caused Xax to leap up in the air. A second tank came rumbling onto the field. Both tanks started shooting with Vivian and the lizards trapped between them.

We will write a program that shows the tanks firing at Vivian and the lizards, who are trapped between them.

The sound effects we use will be similar to those in some video games. First we will draw the tanks. The tanks will appear to come in from opposite sides of the screen. The FOR/NEXT loop of lines 30 through 120 causes this effect. Now enter and RUN the following program:

```

10 PRINT "SHIFT CLR/HOME"
20 FOR A = 1 TO 22 : PRINT : NEXT A
30 FOR T = 1 TO 7
40   FOR A = 0 TO 2
50     PRINT TAB(T) ; "  ☞  "; TAB(39-T) ; "  ☞  ";
60     IF A <> 1 THEN GOTO 85
70     PRINT : PRINT "CRSR ↑";
80     PRINT TAB(T + 1) ; "  ☞  "; TAB(38-T) ; "  ☞  ";
90     IF A = 2 THEN PRINT "CRSR ↑[3X]";
100    PRINT
110  NEXT A
120 NEXT T
500 GOTO 500

```

This program uses a common programming technique called *branching*, as in the branches of a tree. We want the middle line of the tank to extend one column beyond the others to represent the gun barrel. The

“branching” statement at line 65 tells the computer to avoid lines 70 and 75, which draw the extra character, unless $A = 1$.

Now we will have the tanks shoot toward the middle of the field where Vivian and the lizard creatures are located. A bullet is just the graphics character **SHIFT Q** at a particular location. To have a bullet move we plot it and then erase it by plotting it at the same location using a blank space. Then we plot it again slightly further along in its trajectory, then erase it, then plot it again, and so on. The sequence of drawing and erasing is done so fast by the computer that what you see appears to be bullets moving. Now add the following lines and RUN the revised program.

This program will soon be too long to be listed on one screen. If you want to review part of the program, say lines 100 to 200, type LIST 100-200 and press **RETURN**. As usual the underline symbols denote blank spaces.

```
125 PRINT
127 FOR D = 11 TO 19
130 FOR V = 9 TO D
140 PRINT "CRSR↑"; : PRINT
145 IF V = D THEN PRINT
```

```
150 PRINT TAB (V) ; " SHIFT Q"; TAB (39-V) ; " SHIFT Q";  
155 FOR T = 1 TO 50: NEXT T  
160 PRINT "CRSR↑"; : PRINT  
165 PRINT TAB (V) ; " _ "; TAB (39-V) ; " _ ";  
170 NEXT V  
175 FOR T = 1 TO 200: NEXT T  
177 PRINT "CRSR↑CRSR↑"; : PRINT  
180 NEXT D
```

Before we add sound, add this line and RUN the program:

```
137 PRINT " CTRL 1";
```

The bullets are now black! If you stop the program using only the **RUN/STOP** key, the computer continues to print in black. If you also use the **RESTORE** key, it goes back to the usual color.

Pressing **CTRL** and any key from 1-8, or **C** and 1-8, instructs the computer to print in one of 16 colors. (See the glossary for a table of colors.) To insert a color change in a program, just put a line in the program that says

```
PRINT " [COLOR INSTRUCTIONS] ";
```

For instance, add

```
40 PRINT "CTRL 6";
```

to our tank program. Of course, the same could be done to any of the earlier programs.

As we add sound, we will explain the POKEs in more detail. First, we set the volume at maximum level.

```
12 POKE 54296, 15
```

If you were to run the program now, you would hear a little “click” as the speaker in the TV is tugged by line 12. Unless you use the **RESTORE** key, you won’t hear the click when you run it again, as the volume setting remains at 15.

To get a pitch, we have to POKE values into both the “HIGH FREQUENCY” address 54273 and the “LOW FREQUENCY” address 54272. Both accept values from 1 to 255. The basic idea is the higher the value, the higher the note, with 54273 making the “gross adjustment” and 54272 the “fine tuning.”

```
14 POKE 54273, 10 : POKE 54272, 17
```

In order to hear the pitch, we must POKE a value into the SUSTAIN/RELEASE value at address 54278. 248 is a high value. Zero, the initial value, is inaudible.

```
16 POKE 54278, 248
```

Finally, we activate VOICE 1 with the “noise” setting (or “waveform”) at address 54276

```
18 POKE 54276, 129
```

RUN the program. Now we hear the roar of the tank. To get the explosion of the tanks firing add these lines:

```
128 POKE 54276, 129: POKE 54296, 15
```

```
157 POKE 54296, 19-V
```

```
172 POKE 54276, 128
```

Line 128 sets the waveform at “noise” and the volume at maximum at the moment of “firing.”

Line 157 decreases the volume for an “echo” effect, using the position variable V.

Line 172 shuts off the note at VOICE 1 at the moment when the bullets disappear.

Finally, we should see what happened to the two tanks. Enter the following lines and RUN the finished program. Remember that the underline symbols denote blank spaces.



```
190 PRINT "CRSR↑";  
195 POKE 54276,129 : POKE 54296,15  
200 FOR X = 1 TO 120 : PRINT "_ "; : NEXT X  
210 FOR T = 14 TO 0 STEP -1  
220 POKE 54296,T : FOR Z = 1 TO 25: NEXT Z  
230 NEXT T
```

Note that in line 210 we are counting “backwards.”

Suddenly, the ground gave way and both tanks sank out of sight into the mud. The field was wet from recent rains and couldn't support the weight of the tanks. Vivian and her companions cheered. “Move fast before more of them come!” shouted Xavier as he and his followers scurried toward the woods.

Soon they came to another clearing. An old farmhouse was at the far end of the clearing. A propeller car was parked in front of the house. Three xamsters grazed in a field next to the house. A group of small chicken-like birds with purple and green feathers were squawking and hopping around.

As Vivian approached the house she saw a lizard sitting in a chair on the front porch. He was wearing a large hat and a piece of straw was hanging out of the corner of his mouth.



He reminded Vivian of her Uncle John, who was a farmer in California.

The lizard farmer saw the visitors and stood up, squinting at them. "Xaviar!" he said when he recognized Xaviar, "I haven't seen you in many a Yadnom. What brings you here? Isn't it dangerous for you to come down from the mountains? If the forces of the Leadership Council find you here they will put you in jail or do something even worse."

"Greetings, Xim," said Xaviar. "I'm here because my wish to become ruler of Planet X is finally going to come true. We just evaded a group of soldiers from the Leadership Council. Just as I wished, Vivian has come from outer space to help us find fuel. We need to go to her spaceship so she can use her equipment. May I borrow your car?"

"I reckon so," said Xim. "I don't have any fuel though. You'll have to hitch it to a xamster."

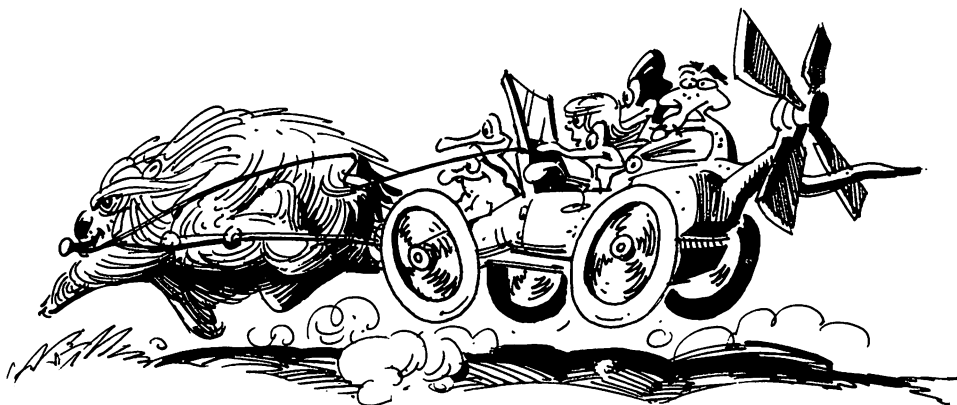
"Fine," said Xaviar, who was already walking toward one of the grazing xamsters.

Soon Xim's propeller car was hitched up to a xamster. "Whoa, boy," said Xim, as he calmed down the protesting beast.

"I'll return your car as soon as possible," said Xaviar.

"Take your time," said Xim. "I wasn't planning to go anywhere."

Xax, Vivian, and Xaviar squeezed into the front seat of the propeller car and the other lizards climbed onto the hood and trunk. "Giddyap!" shouted Xaviar as he snapped the reins.



The hamster reared back and suddenly lurched forward. Vivian fell back in her seat and Xete toppled off the side of the propeller car into the bushes. Xete quickly brushed himself off and climbed back on the car and the hamster pulled the car down the last hill to the city. "We'll go around the outskirts of town to get to the road that leads to the spaceship," said Xavier.

Buildings came into sight. Vivian and the lizards entered the outskirts of the city.

We will write a program that shows the buildings of the city as Vivian saw them when she and her lizard companions drove down the hill. This program illustrates the computer's ability to quickly and accurately perform a set of instructions over and over. This is one of the reasons why computers are so useful. The computer will draw a number of buildings, accompanied by building noise.

Clear the computer's memory. Then enter and RUN the following program:

```
500 FOR V = 1 TO 3
510 PRINT "G + [3X] CRSR←[3X] CRSR ↓";
```

```
520 PRINT "█ + _ █ + CRSR⇐[3X] CRSR↓";  
530 NEXT V
```

This will be our "building" subroutine. To get a row of buildings, add these lines:

```
20 FOR W = 1 TO 7  
30 GOSUB 500  
40 PRINT "CRSR↑[6X] CRSR⇒[6X] ";  
50 NEXT W  
400 GOTO 400  
540 RETURN
```

For three rows, add:

```
10 FOR Q = 1 TO 3  
60 PRINT "CRSR⇐[2X] CRSR↓[7X] ";  
70 NEXT Q
```

Of course, you can color these buildings using color controls. For instance,

```
5 PRINT " CTRL 3";
```

To add building sounds, enter these lines

```
1 POKE 54296, 15 : POKE 54278, 248  
2 POKE 54273, 34 : POKE 54272, 75
```

```
3 POKE 54276, 129
502 POKE 54296, 15
505 POKE 54273, 34 + 10*Q*V
532 FOR X = 1 TO 40: NEXT X
535 POKE 54296, 0
```

Line 505 raises the pitch of the sound as Q and V increase.

A low rumbling sound seemed to be coming from the direction they were headed. Growing nearer, Vivian recognized the sounds of music and marching and singing. Around a corner, Vivian and the lizards met a large group of lizard creatures marching through the streets. They were protesting against the Leadership Council and demanding more fuel.

BACK TO THE SPACESHIP

The protesters marched through the streets of the city demanding more fuel and a new government. "MORE ORE! MORE ORE!" they chanted, referring to the purple metal ore used to make fuel for Planet X machines. Vivian and the lizards watched from their propeller car as the marchers came down a street and stopped. A group of lizards with instruments were playing a strange melody. The marchers stopped in front of a funny looking lizard creature who had bulging eyes and looked something like a chameleon. He stood on top of a parked propeller car and was giving a speech.

"Hear me, fellow citizens!" said the lizard with the bulging eyes. "We are here today to protest the policies of a government that can't even find fuel to run the cars of the people!"

"Hear! Hear!" shouted the protesters.

"We have a government that puts its citizens in jail for suggesting new ways to do things... A government that has so many rules that citizens have made careers trying to figure them out!"

"Hear! Hear!" shouted the protesters.

"You see," said Xavier to Vivian, "they want a new leader ... Someone who will help Planet X make progress."

"We have a Leadership Council that won't listen to anyone who wants to improve our planet," continued the lizard with the bulging eyes.



"I'll listen!" said Xaviar to Vivian.

"A Leadership Council that can't agree on what to do about anything!" said the lizard with the bulging eyes.

"I know what to do about everything!" said Xaviar.

"A Leadership Council made up of fools, not lizards!" said the lizard with the bulging eyes.

"I'm a lizard, not a fool!" said the excited Xaviar.

"We need a leader, not a Leadership Council!" shouted the lizard with the bulging eyes.

"And I am that leader!" said Xaviar to Vivian.

"And I am that leader!" shouted the lizard with the bulging eyes to the crowd.

"What?!" exclaimed Xaviar. "He can't do that! I'm going to be the next leader of Planet X."

"Hear! Hear!" shouted the crowd.

"Shh," said Vivian to Xaviar. "If the wrong lizards see you, they'll come get you. Anyway, we have to go to my spaceship. Once we find fuel the citizens will choose you to be their leader."

"I will be your leader!" said the lizard with the bulging eyes to the crowd.

"Hear! Hear!" shouted the crowd.

"You're right," said Xaviar to Vivian. "Once I find fuel, they'll want me to be leader. Let's go! Giddyap!"

The xamster, who had fallen asleep during Bulging Eye's speech, opened his eyes, looked around, and started trotting down the road.

We will write a program that illustrates the protest march. First we will do the protest music, for which we must have a more complete discussion of sound.

The “sound chip” (locations 54272 through 54296 in the computer’s memory) behaves a little differently from other parts of the computer in that it must be “awakened” before it will work. Enter and RUN the following program, which we will explain line by line.

```
10 FOR L = 54272 TO 54296 : POKE L, 0 : NEXT L
```

This “clears” the sound chip and readies it for action.

```
20 V = 54296 : H = 54273 : L = 54272 : W = 54276
```

These “abbreviations” save us the trouble of typing five-digit numbers repeatedly. V is the volume address, H and L are high and low frequency pitches for VOICE 1, and W is the waveform for VOICE 1.

```
30 S = 54278 : A = 54277
```

S is “SUSTAIN/RELEASE” for VOICE 1. A is “ATTACK/DECAY.” We haven’t really gone into these yet.

40 POKE V, 15

This sets the volume at 15 (maximum).

50 POKE H, 15 : POKE L, 37

This enters pitch values for VOICE 1.

60 POKE S, 240

Sets a high "SUSTAIN" value.

70 POKE W, 17

Sets the waveform at "triangle."

Now RUN the program.

The pitch that you hear will continue until you either press **RUN/STOP**, **RESTORE** or type POKE V,0 and press **RETURN**.

Address S is the "SUSTAIN/RELEASE" value. It works something like the pedals on a piano. Entering the numbers 128,64,32,16 or any sum of two to four of these numbers (such as $128 + 64 + 32 + 16 = 240$) sustains the note at a percent of the volume in location V. If you don't POKE a value into S it reads 0 and you get a zero sustain value.

Add these lines and RUN the program.

```
45 FOR P = 1 TO 255
50 POKE H, P : POKE L, 37
75 NEXT P
80 POKE W, 16
```

Lines 45, 50, and 75 form a loop which takes the high frequency value at H from 1 (lowest) to 255 (highest). 80 shuts off the triangle waveform.

The other waveforms available are “sawtooth,” “square,” and “noise.” To hear “sawtooth” substitute:

```
70 POKE W, 33
80 POKE W, 32
```

For “noise” substitute:

```
70 POKE W, 129
80 POKE W, 128
```

You can’t hear “square” from this program because it requires two additional “POKEs.” We are not going to use this waveform in the protest music but if you want to hear it, substitute these lines:

```
35 POKE 54275, 15 : POKE 54274, 15
70 POKE W, 65
80 POKE W, 64
```

To hear the effect of the low frequency, we take a fixed value of high frequency (using the triangle waveform). Make these changes:

```
35
```

(This erases line 35 if you used it to get the "square" wave.)

```
50 POKE H, 17 : POKE L, P
70 POKE W, 17
80 POKE W, 16
```

You should hear a very slight change in pitch like a guitar player fine-tuning a string.

We will now do a "sequence" of notes. (Values for high and low frequencies are taken from the table in the back of the book.) Line 72 gives the notes a length in time.

```
45 FOR P = 7 TO 4 STEP -1
50 POKE H, P : POKE L, 255
```

```
72 FOR T = 1 TO 250 : NEXT T
90 GOTO 45
```

This is an OK “bass line,” but the notes have no “shape.” We will give them some using the “ATTACK-DECAY” and “SUSTAIN/RELEASE” values. Make these changes:

```
47 POKE A, 15
60 POKE S, 248
72 FOR T = 1 TO 200: NEXT T
75
85 FOR T = 1 TO 50: NEXT T
87 NEXT P
```

All we are going to say about this now is that address A affects what happens at the beginning of the note and S affects the middle and end.

To finish the music, we give it a “back beat,” playing “noise” on VOICE 2 when the bass note is not playing. Add these lines and RUN the program:

```
51 POKE 54280, 17 : POKE 54279, 17
(high and low frequency notes for VOICE 2)
52 POKE 54285, 32
(sustain for VOICE 2)
```

```
55 POKE 54283, 128
```

(turns OFF the noise waveform)

```
82 POKE 54283, 129
```

(turns ON the noise waveform)

We shall use the graphics character (**SHIFT** X) for the marchers. To get the slogan "MORE ORE" in time with the music, add the following lines.

```
1 PRINT " SHIFT CLR/HOME ";
2 PRINT "CRSR↓[10X] ";
46 GOSUB 525
525 IF INT(D/2) = D/2 THEN A$ = "MORE"
530 IF INT(D/2) <> D/2 THEN A$ = "_ORE"
535 PRINT TAB(18); A$;
540 D = D+1 : PRINT "CRSR↑"; : PRINT
550 RETURN
```

To add marchers, add these lines. In line 510, type
_ **SHIFT** X _ twelve times within the quotes

```
46 GOSUB 500
500 PRINT: PRINT: PRINT
510 PRINT TAB(7-P); "_ SHIFT X_ "; [12X]
515 PRINT
520 PRINT "CRSR↑[4X] ";
550 RETURN
```



Vivian and the lizards went around another corner and came upon a contingent of soldier lizards who had been sent by the Leadership Council to monitor the protest march.

"Look!" said one of the soldiers. "It's Xaviar and the girl who escaped from jail!"

"Get them!" said another.

The soldiers started running toward the propeller car. Xaviar yanked on the reins, making the xamster's head jerk sideways. The xamster turned and ran down a side street with the soldiers in pursuit.

"Faster!" shouted Vivian. "If they catch us they'll put us all in jail."

The xamster was able to run faster than the soldiers who were all on foot. A few blocks later, the soldiers were far behind the propeller car.

"We must hurry," said Xaviar. "Soon the army will come in their cars." He snapped the reins and the xamster went faster.

As she rode along the same bumpy dirt road that she had been on when she first came into the town with Xang, Vivian anxiously wondered how she was going to escape from Planet X. She was afraid that the soldiers were going to catch up with them. Even if they didn't, she didn't know how she would get away from Xaviar.

After a long, uncomfortable ride, Vivian and the lizards reached the field where the spaceship had landed. The xamster



pulled the propeller car across the field. Two other propeller cars appeared on a nearby hill. One was pulled by a xamster; the other was powered by its own engine. Both were filled with lizards.

"It's the army of the Leadership Council!" shouted Xon.

"It's the Leadership Council themselves!" shouted Xax.

Vivian looked at the approaching propeller cars. One of them was filled with lizard soldiers. In the other were four lizard creatures wearing pointed hats and dressed in purple robes. This car was driven by Xang. Vivian guessed that this was the Leadership Council.

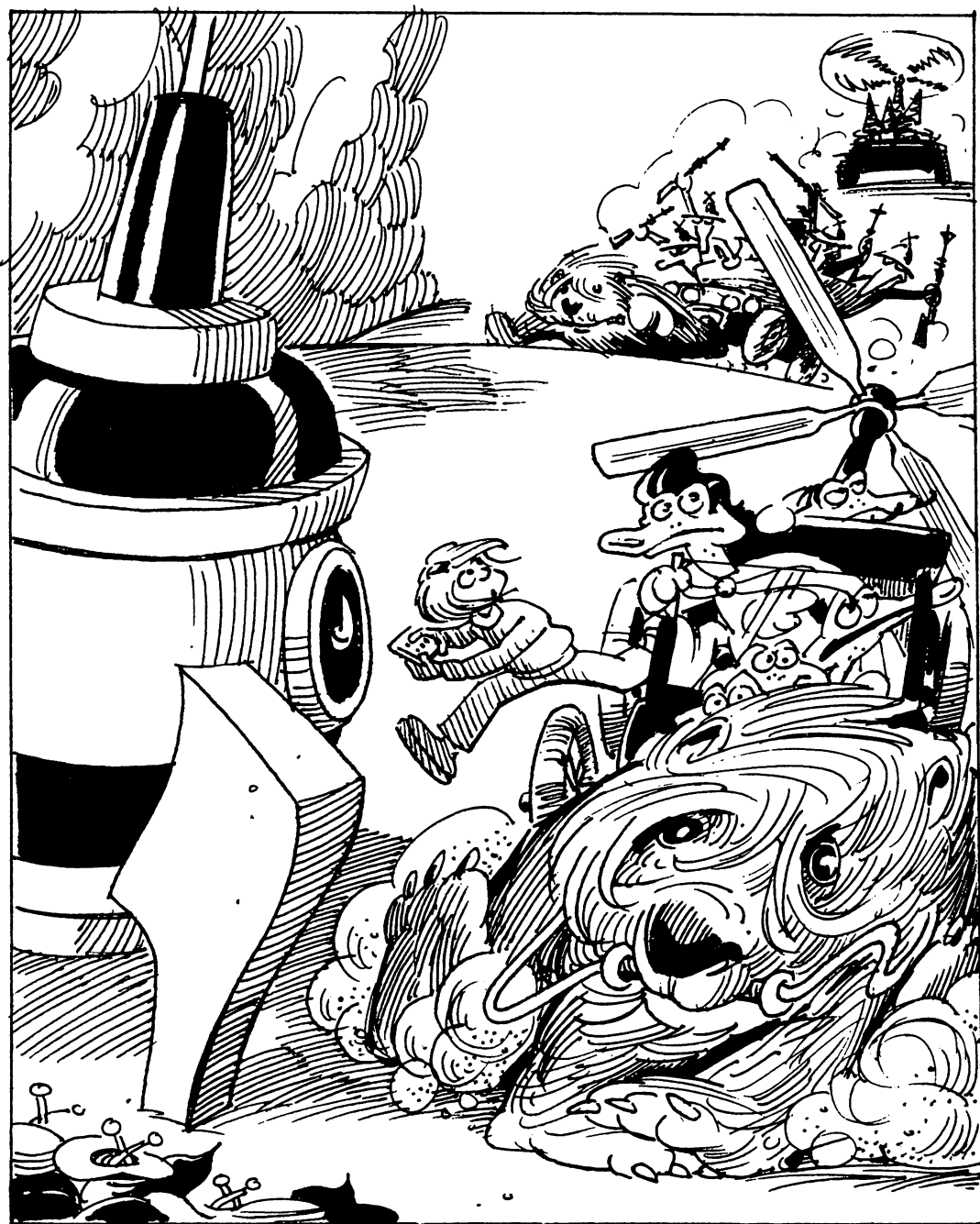
The propeller cars came closer. One of the lizard soldiers boomed in a loud voice, "YOU ARE UNDER ARREST! DO NOT MOVE!"

Vivian realized that if the lizard soldiers put her in jail, she might never get away. She pressed a button on her pocket computer and a ladder descended from her spaceship. She scurried up the ladder and looked down to see Xavier following right behind her. Then the lizards raised their weapons.

"STOP OR WE'LL SHOOT!" said a lizard soldier.

"Don't shoot!" said Xavier. "This girl has come from outer space. She is going to find fuel for us. Let us go and search for fuel."

One of the robed lizards slowly stood up. He looked older than the other lizards. He motioned for silence before he



solemnly spoke. "Xaviar," said the old lizard in a hoarse voice, "when your father and I were young we used to hunt crickets together. We hid under rocks in a stream that long ago dried up. We played together in a forest where a city now stands. Life was different then. Food was plentiful and so was fuel. Your father and I didn't have a care in the world. Now things have changed. Our fuel supply has dwindled to practically nothing. Our soil is poor. This is a time of hunger and scarcity."

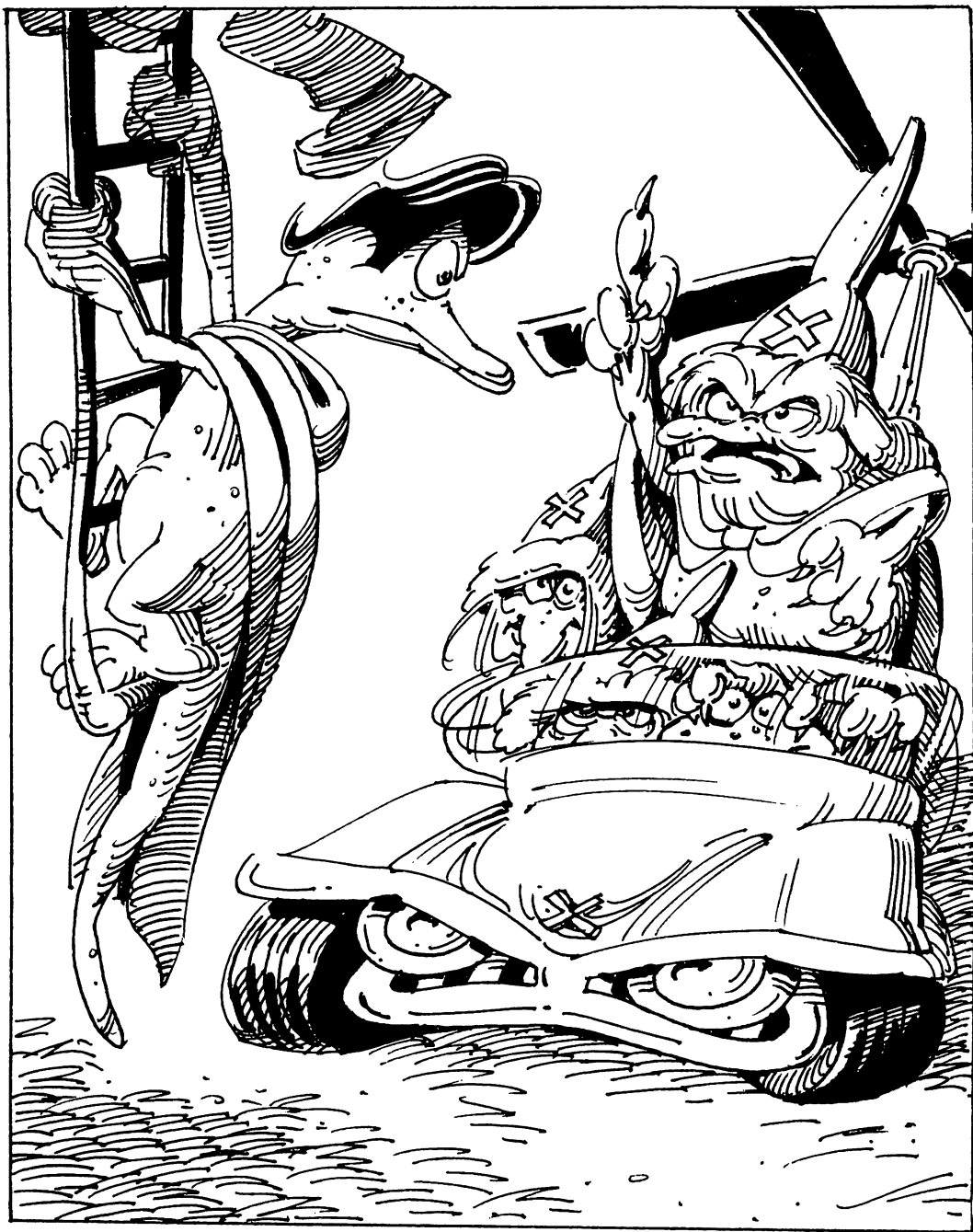
"You were a rebellious youth," the old lizard continued. "Your father never knew what to do with you. You were his pride and joy. 'Xaviar is so smart,' he used to say, 'but he doesn't like anybody to tell him what to do.'"

The old lizard paused to clear his throat. "Now look at yourself," he continued. "Leader of the Rebel Forces. Xaviar, it's time to change. Join us on the Leadership Council. You're a bright boy, Xaviar, and the citizens like you. Join us now and we won't throw any of your followers in jail."

"But Vivian and I will find fuel," said Xaviar.

"We'll find our own fuel," said the old lizard. "We don't know where this girl is from or why she is here. Who knows what she is planning? Come and join us."

While the old lizard was talking to Xaviar, Vivian was quietly climbing up the ladder.



"The girl has come here because I wished that she would," said Xaviar. "The trouble with you is that you never want to do anything new. You won't listen to new ideas or think about things in different ways. You're tired and feeble, old lizard, and the citizens are angry. They want someone else to be their leader and that someone will be me. I'm going with Vivian to find fuel and when I come back I will be the new leader of Planet X."

Vivian had just reached the door of her spaceship when Xaviar finished talking. She opened the door and tried to climb inside the spaceship before Xaviar got up the ladder, but Xaviar was too fast.

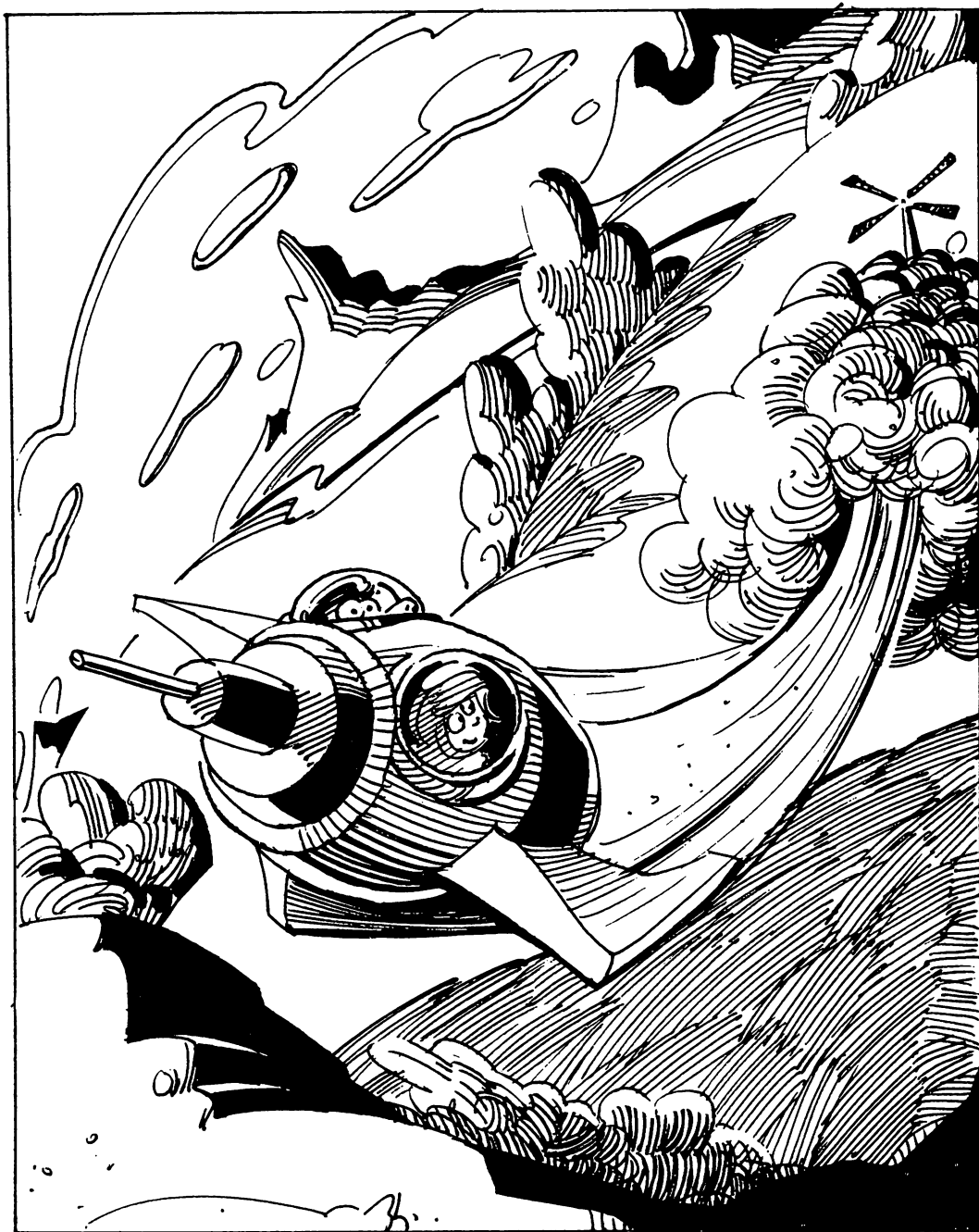
"STOP!" ordered the old lizard, but Xaviar had already climbed into the spaceship and closed the door.

"Please, Xaviar," said Vivian inside the spaceship, "go back and be on the Leadership Council. They need you to help them and I need to go home."

"Don't be ridiculous," said Xaviar. "I'll go back and you can go home, but first we must find fuel."

Vivian saw that there was no way she could get Xaviar to leave the spaceship without first looking for fuel. "Very well," she said. "I'll circle the planet and use my laser scanner. But then I must be on my way. School starts in a few weeks and I can't be late. Sit there and fasten your seatbelt."

Xaviar and Vivian sat down and fastened their seatbelts. Vivian turned on the engines. With a mighty roar, her spaceship rose into the sky. The lizards below watched in amazement.



We will write a program that shows Vivian's spaceship blast off from Planet X. First we will show an effective way to put graphics on the screen. Enter and RUN the following program:

```
10 PRINT " SHIFT CLR/HOME "  
20 POKE 55296+20+40*12, 2  
30 FOR N = 0 TO 255  
40 POKE 1024+20+40*12, N  
50 FOR T = 1 TO 250 : NEXT T  
60 NEXT N
```

It is possible, using POKE statements, to place any graphics character anywhere on the screen in any of the 16 colors, or a character in a reverse field (background color and character color reversed). This reversal is available directly: press **CTRL** 9 and type something. To eliminate the reversal, press **CTRL** 0. There are 25 rows and 40 columns, or $25 \times 40 = 1000$ locations available on the screen. To have a character appear in one of the 16 available colors (see Glossary) at the point in column M and row N, use the statement

```
POKE 55296+M+40*N, (number of color)
```

For instance, statement 20 above applies RED at column 20, row 12. Each graphics character has a number associated with it. For instance, the character  has the number 105 as its "code." To put this character in column M, row N, use the statement

POKE 1024+M+40*N, 105

Try this with $M = 20$, $N = 12$. In combination with the previous POKE, the character should appear in red. When the spaceship program is finished, it will be the longest program in the book. We will build it in stages. The first part of the program shows the spaceship before take-off. The shape of the spaceship is irregular, so we will use several different graphics characters. Clear the computer's memory, then enter and RUN the following program:

```
10 PRINT "   "  
20 FOR N = 0 TO 24  
30 FOR M = 16 TO 23  
40 POKE 55296+M+40*N, 7  
50 NEXT M: NEXT N  
60 FOR N = 13 TO 22
```

```
70 FOR M = 18 TO 21
80 POKE 1024+M+40*N, 160
90 NEXT M : NEXT N
```

Lines 20-50 set the color for the whole “corridor” we are using, which takes a little time. Lines 60-90 draw the body. (RUN the program and see.) Next we write a subroutine to draw the nose:

```
100 F = 12 : GOSUB 500 : END
500 POKE 1024+19+40*(F-2) , 233
510 POKE 1024+20+40*(F-2) , 223
520 POKE 1024+18+40*(F-1) , 233
530 POKE 1024+19+40*(F-1) , 160
535 POKE 1024+20+40*(F-1) , 160
540 POKE 1024+21+40*(F-1) , 223
550 FOR T = 18 TO 21
560 POKE 1024+T+40*F, 160 : NEXT T
580 RETURN
```

RUN the program. Now we will write a subroutine to draw the tail section:

```
100 F = 12 : GOSUB 500 : GOSUB 600
400 GOTO 400
```



```
600 FOR T = 17 TO 22: Y = 160
620 IF T = 17 THEN Y = 233
630 IF T = 22 THEN Y = 223
640 POKE 1024+T+40*(F+11), Y: NEXT T
660 FOR T = 16 TO 23: Y = 160
680 IF T = 16 THEN Y = 233
690 IF T = 23 THEN Y = 223
700 POKE 1024+T+40*(F+12), Y: NEXT T
710 RETURN
```

RUN the program.

Now we will have the spaceship blast off. Enter these lines and RUN the program:

```
110 FOR F = 11 TO 2 STEP -1
120 GOSUB 500 : GOSUB 600 : GOSUB 800
150 NEXT F
800 FOR X = 16 TO 23
810 POKE 1024+X+40*(F+13), 32: NEXT X
820 RETURN
```

Subroutine 800 erases the bottom line of the ship after the loop of 110-150 has redrawn the ship one row higher. We will now get the nose to vanish:

```
160 F = 1: GOSUB 520 : GOSUB 600 : GOSUB 800
170 F = 0: GOSUB 550 : GOSUB 600 : GOSUB 800
```

We now get the rest of the body of the ship to vanish:

```
180 FOR F = -1 TO -11 STEP -1
190 GOSUB 600 : GOSUB 800 : NEXT F
```

Now for the tail section:

```
200 F = -12 : GOSUB 660 : GOSUB 800
210 F = -13 : GOSUB 800
```

Now for the sound of the ship:

```
11 FOR L = 54272 TO 54296 : POKE L, 0: NEXT L
12 POKE 54296, 15 : POKE 54277, 240
14 POKE 54273, 17 : POKE 54272, 37
16 POKE 54278, 255 : POKE 54276, 129
220 POKE 54276, 128
```

You can really hear the effect of the “ATTACK-DECAY” value poked into 54277 as the sound slowly comes up (slow attack) and the “SUSTAIN-RELEASE” value in 54278 (high sustain, long release). The “release” begins after the waveform has been turned off (line 220).

ANOTHER CALL FOR HELP

Vivian was glad to blast off into space, even if Xavier was with her. When the spaceship was high enough, she leveled off and started flying over Planet X. They flew across a valley and above a mountain range. Vivian turned on her laser scanner and sent a beam across the planet's surface. If there were underground deposits of fuel, the laser beam would cause a signal to appear on the computer screen.

"I never knew that Planet X looked like this!" exclaimed Xavier as he peered through the viewport.

Vivian was busy scanning the planet for fuel. The spaceship passed over a mountain range, a lake, a forest, and rolling hills. Each time a new scene came into view, Xavier gasped in amazement.

"A city! Look at the tops of the buildings," he said.

Vivian circled the city a few times so Xavier could see it clearly.

Vivian watched the Monday sun sinking lower in the sky. Or was it the Yadnom sun? In any event, she wished that Xavier would stop distracting her so she could concentrate on finding fuel.

We will write two programs. The first program shows the laser scanner display. Lines 70 to 110 draw the screen rectangle. Line 500 keeps the computer from

printing READY. Now enter and RUN the following program:

```
10 PRINT " SHIFT CLR/HOME "  
20 FOR N = 6 TO 16 STEP 10  
30 FOR M = 1 TO 37 STEP 3  
40 POKE 55296+M+40*N, 7  
50 POKE 1024+M+40*N, 160  
60 NEXT M: NEXT N  
70 FOR N = 8 TO 14  
80 FOR M = 7 TO 31: Y = 0  
83 IF N = 8 OR N = 14 THEN Y = 10  
86 IF M = 7 OR M = 31 THEN Y = 10  
90 POKE 55296+M+40*N, Y  
100 POKE 1024+M+40*N, 160  
110 NEXT M: NEXT N  
500 GOTO 500
```

We mentioned that there are three special variables reserved for special use: ST, TI, and TI\$. ST is used in connection with loading programs from disk or tape, and we will not deal with it. The variable TI has value 0 when the computer is turned on and is increased by 1 every 1/60th of a second, so it is a sort of "clock." TI\$ is

a six-place string of numerical symbols which represent the hour, minutes, and seconds of a real time clock. For example,

083022

represents 8:30 A.M. plus 22 seconds. When you turn on your Commodore 64, this clock is set at 000000. To set it at the actual time, type TI\$ = "[actual time]" and press **RETURN**. Do this and put the real-time clock into the program as follows:

```
500 FOR X = 0 TO 22:PRINT: NEXT X
505 PRINT " CTRL 9";
510 PRINT TAB(16) ; TI$;
520 PRINT "CRSR←[6X]"; : GOTO 510
```

Line 505 makes the computer print "in reverse."

The next program will show the tops of the buildings as seen by Xaviar from the viewport as the spaceship passed over them. Type NEW to erase the previous program. Then enter and RUN the following program:

```
10 POKE 53280,0:POKE 53281,0:PRINT" SHIFT CLR/HOME "
100 FOR M = 0 TO 30 STEP 10
110 FOR L = 0 TO 5
```

```
120 POKE 55296+M+L+40*24, 15
130 POKE 1024+M+L+40*24, 160
140 NEXT L: NEXT M
```

This draws the tops of the buildings.

```
150 FOR X = 0 TO 5
160 FOR M = 0 TO 30 STEP 10
170 POKE 55296+M+5+X+40*24, 15
180 POKE 1024+M+5+X+40*24, 160
190 POKE 55296+M+X+40*24, 0
200 NEXT M: NEXT X
```

This moves the buildings across the screen.

```
210 FOR X = 0 TO 5
220 FOR M = 4 TO 34 STEP 10
230 POKE 55296+M-4+X+40*24, 15
240 POKE 1024+M-4+X+40*24, 160
250 POKE 55296+M+X+40*24, 0
260 NEXT M: NEXT X
270 GOTO 150
```

This moves the right-hand building off-screen and brings a new one in on the left until the original position recurs.

The amount of typing in this program could be lessened by writing a subroutine that would handle lines 150 through 200 and lines 210 through 260. You might try to do this.

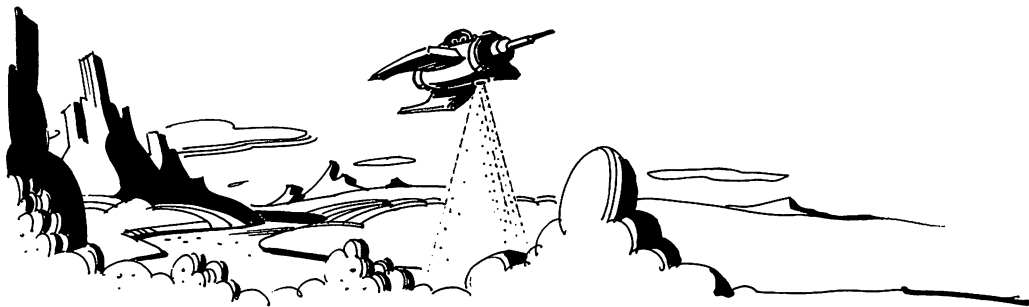
Let's add sound and music to the program. For the sound of the engine, add

```
5 FOR L = 54272 TO 54296: POKE L, 0: NEXT L
20 POKE 54296, 15: POKE 54278, 128
25 POKE 54273, 17: POKE 54272, 37
30 POKE 54276, 129
```

and RUN the program.

Next let's add a high, "science-fiction" pattern in VOICE 2:

```
155 GOSUB 500
215 GOSUB 500
500 READ A, B
510 POKE 54280, A: POKE 54279, B
520 POKE 54284, 136: POKE 54285, 128
530 POKE 54296, 15: POKE 54283, 17
535 DATA 168, 250, 142, 24, 179, 6, 134, 30
540 IF B = 30 THEN RESTORE
550 RETURN
```

Now we'll add a low line in VOICE 3:

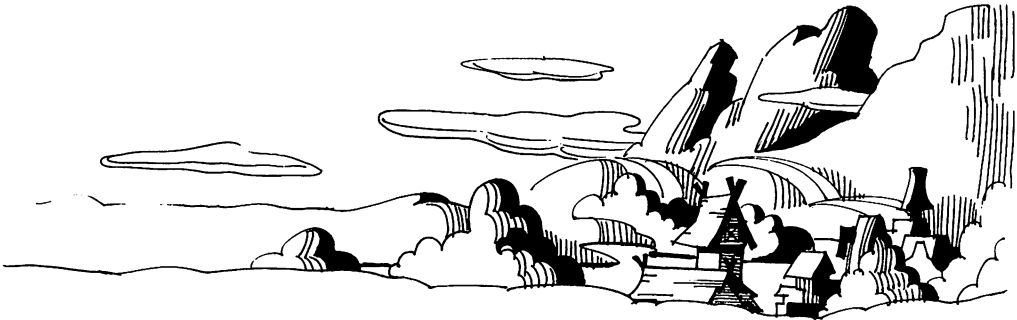
```
156 C = 7: D = 12: GOSUB 600
205 C = 9: D = 247: GOSUB 600
600 POKE 54291, 24: POKE 54292, 200
610 POKE 54287, C: POKE 54286, D
620 POKE 54290, 33
630 RETURN
```

For another note and a "hold" in the flight plan, add these lines:

```
105 GOSUB 500
107 C = 14: D = 24: GOSUB 600
270 GOTO 100
```

Finally we add a routine which randomly twinkles stars in the Planet X sky:

```
106 GOSUB 700
158 GOSUB 700
217 GOSUB 700
700 E = INT(RND(1)*40): F = INT(RND(1)*21)
710 POKE 55296+E+40*F, 1
720 POKE 1024+E+40*F, 42
730 POKE 55296+E+40*F, 0: RETURN
```



The tempo of the music has changed because the computer now has more to do between each change of note. This is a factor in accompanying video images with music in BASIC, which is a "slow" language.

Scanning the countryside with her laser scanner, Vivian wondered if the citizens of Planet X would want Xavier to be their leader. She hoped that Xavier would let her go home when they returned from their mission.

Vivian was daydreaming and listening to Xavier describe the view when the scanner display lit up with a series of red blips. Vivian focused the scanner on the source of the blips and discovered that an underground fuel deposit was located at the base of a nearby mountain next to a lake. Further scanning indicated that the deposit was a large one.

"Xaviar!" said Vivian, as she maneuvered the ship toward the mountain. "I've found fuel!"

"Great!" shouted Xavier as he jumped out of his seat and ran over to look at the laser scanner. "Where?" he asked.

"There is a large underground ore deposit at the base of the mountain," said Vivian, pointing to the blips on her screen. "There may be other deposits nearby."

Xaviar shook Vivian's hand. "We did it!" he exclaimed. "We found fuel! We saved the ... "

Suddenly the spaceship started violently shaking back and forth.

"What's going on?" said Xavier as he held onto the instrument panel to keep his balance.

"I don't know," said Vivian, who was frantically trying to get to the computer. "There is some kind of air turbulence. If it doesn't stop we could be blown into the side of the mountain."

"Try to land," shouted Xavier, "before the spaceship gets blown apart!"

It seemed as though the spaceship would indeed get blown apart if the shaking didn't stop. Vivian had never experienced such violent turbulence, not even in meteor storms or space quakes, and did not know what to do. Holding on to whatever she could, she made her way across the room to the computer controls. She was stunned by what she saw, for the control screens were flashing on and off in a series of random patterns.

"Something weird is happening!" shouted Vivian to Xavier above the din. "The computer has broken down... I've lost control of the ship. I can't get a reading on our altitude or how fast we are traveling... or anything!"

The spaceship shook violently and started to spin. Xavier saw huge rocks and uprooted trees fly by the viewport. The spaceship flipped upside down so the viewport was pointing toward the sky. "Look!" shouted Xavier. "Both suns are in the sky! It's the Monday-Yadnom Intersect!"

From her place at the computer control panel, Vivian could see the two suns through the viewport. "What?" shouted Vivian, who was sure the spaceship would blow apart at any minute.

"The Monday-Yadnom intersect — It's when both suns cross the sky at the same time," shouted Xaviar. "It hasn't happened in my lifetime. I've read about it in books. Strange things take place but I can't exactly remember what."

The spaceship rocked up and down like a car going along an unpaved road. Vivian tried to steer the spaceship but the controls didn't function.

"Look at my watch!" shouted Xaviar. The numbers on Xaviar's Monday-Yadnom digital watch were flashing on and off. "Time is out of control!" shouted the lizard. "I remember now! In a Monday-Yadnom intersect time ceases to exist."

Xaviar could say no more, for the spaceship now started spinning around as if it were caught in an enormous tornado. As they spun, Vivian thought of her parents and her friends on Earth and of how she could have been safely at home babysitting for little Harold Bloomgarden, who lived down the street, instead of being caught with a strange lizard in a world where time made no sense.

The spaceship spun faster and Vivian found herself forced against the wall. She couldn't remember where she was or what she was doing. With an enormous explosion, everything disappeared in a blur of light and sound. Then there was silence.



Enter and RUN the following program, which illustrates the enormous explosion on Planet X.

```
5 FOR L = 54272 TO 54296: POKE L, 0: NEXT L
10 FOR B = 0 TO 15
20 POKE 54276, 128
30 POKE 54296, 15: POKE 54277, 30
40 POKE 54273, 17: POKE 54272, 37
50 POKE 54276, 129
90 PRINT " SHIFT CLR/HOME ": POKE 53280, B: POKE 53281, B
100 FOR F = 1 TO 70
110 C = INT (RND (1) *16)
120 M = INT (RND (1) *40) : N = INT (RND (1) *25)
130 POKE 55296+M+40*N, C
140 POKE 1024+M+40*N, 160
150 NEXT F
160 NEXT B
170 GOTO 10
```

Later, Vivian wasn't sure whether she had been dreaming or not. She remembered that the spaceship had stopped bouncing and shaking and that there was quiet, calm darkness. She felt that there was no hurry to do anything, as if

time had stopped. It seemed to Vivian that she could sit happily forever in the purple darkness of her spaceship.

Slowly, light returned. Vivian looked around the spaceship and saw Xavier sitting on the floor near her. At first she thought he was asleep or even dead but then he looked at her.

Vivian checked the instrument panel and saw that everything seemed to be working normally again. Looking through the viewport, she saw that they were hovering above the lake near where the fuel deposit had been discovered.

"What happened?" said Vivian.

"I'm not sure," said Xavier. "We started spinning and there was an explosion. That's all I remember."

Vivian went to the control panel and turned on the forward thrust rockets. The spaceship moved forward. Everything was normal again.

"I hope you use your new fuel wisely," said Vivian as she and Xavier headed back to the field where they had left the other lizards. "Planet X needs a leader who will respect the environment and be sensitive to the needs of the citizens."

"Don't worry," said Xavier. "once we get the machines running again there won't be any problems. They'll love me."

"They won't love you if you run out of fuel," said Vivian, who had doubts about whether Xavier would be a responsible leader.

The spaceship came back to the field where they had taken off. As Vivian landed the craft, Xavier looked through the viewport. With a blast of the stabilizer rockets, the spaceship came to a rest.

Vivian opened the door of the spaceship and the rays from a Planet X sun greeted her. Although Vivian wasn't sure which one it was, there was now only one sun in the sky.

"Well Xavier, it's been nice knowing you," said Vivian as she shook the lizard leader's hand.

"You have made all of my wishes come true," said Xavier as he gave Vivian a polite lizard kiss. "I do hope you'll come down and say goodbye to everybody."

Xavier stepped through the doorway and onto the ladder. "What happened?!" he shrieked. "Something's wrong!!!"

Vivian looked through the doorway at the scene below. The lizard creatures were still there but they looked different.

Instead of wearing their fashionable lizard clothes, the lizards were wearing primitive garments that looked like caveman clothes. The soldiers were carrying clubs, not rifles. The propeller cars were gone. In their place were xamsters with blankets thrown over their backs. The landscape was different too. There were more trees than before and there was a lake in the middle of the field.

"What happened?" whispered Vivian to Xavier, who was standing next to her in the doorway of the spaceship.

"I think," said Xaviar slowly, "that we went backwards in time. I remember reading a book on Planet X mythology about how once in a while time stops and then reverses itself. I think it is supposed to happen during the Monday-Yadnom Intersect. The two suns appearing together causes this reversal of time. We weren't affected by it because we were in the spaceship. But everything on the ground..."

Then the old lizard who used to be the leader of the Leadership Council stepped forward. He was wearing a robe made of tree bark.

"You from sky," said the old lizard, with a series of grunts. "What you do here?"

"I'm Xaviar... I've returned with fuel," said Xaviar uncertainly, as he stepped out on the ladder.

"Xaviar? Fuel?" said the old lizard curiously. "What means fuel?"

"You know, fuel," said Xaviar with panic in his eyes. "Fuel that is used to make machines run and cars go."

"Machines? Cars? What means machines? What means cars?" said the old lizard, who then began to laugh. Soon all the lizards on the ground were laughing hysterically.

"MACHINES! CARS! MACHINES! CARS!" the lizards chanted. Then the old lizard spoke again.

"Brother from sky," said the old lizard, "come show us how to fly. Tell us about machines and cars and fuel. Lead us."



"LEAD US! LEAD US!" chanted the lizards, as the soldiers pounded the ground with their clubs.

"You must go to them now," said Vivian to Xavier. **"All your wishes have come true,"** she said. **"You have found fuel and they want you to be their leader."**

"But what good is fuel?" said Xavier. **"There are no machines any more. And I don't want to lead lizards who went back in time."**

"You'll be all right. You can teach them about machines and cars and fuel. They'll look up to you," reassured Vivian.

"MACHINES! CARS! FUEL! LEAD US!" chanted the lizards.

"But... But... I want to go with you," said Xavier to Vivian.

"Here," said Vivian. **"Take this to remember me by."** She handed Xavier her pocket computer.

"Oh, thank you," said Xavier. **"But please don't go,"** he said. **"Stay with me on Planet X. We can be rulers together."**

"You know I can't do that," said Vivian. **"Go now and be brave."** She squeezed Xavier's lizard hand and quickly stepped back into the spaceship.

The girl and the lizard took one last look at each other. Vivian pressed a button and closed the door of the spaceship. She watched Xavier descend the ladder to the ground, then retracted the ladder and started the engines. Soon she was blasting off.



Traveling away from Planet X, Vivian thought about Xavlar and the other lizard creatures and her adventures on this strange planet. She was about to put her spaceship on automatic control and take a long nap when she noticed a message appearing on her computer communications screen. She gasped in surprise when she saw "HELP!" appear over and over on the screen.

Vivian discovered that the message was coming from where her spaceship had been just before the explosion. She suddenly realized the source of the message — her own computer!

There was a signal device on Vivian's computer that automatically broadcast a HELP message if the spaceship controls malfunctioned. The message must have been sent when the spaceship was spinning out of control.

Vivian remembered the Monday-Yadnom Intersect and how time had gone backwards. The original HELP message Vivian received that caused her to go to Planet X was, in fact, the same HELP message that she was receiving now. Since they had gone back in time, she had received the message before any of this had ever happened.

Vivian was getting a headache. She didn't want to think about Xavlar or Planet X or HELP messages or going back in time. With a flick of a switch, she turned off her communications computer. Moments later she was fast asleep.

GLOSSARY OF PROGRAMMING TERMS

(For more information consult
the **COMMODORE 64 USER'S GUIDE**)

ARITHMETIC

In Commodore BASIC, the arithmetic operations are represented as follows:

- + addition
- subtraction
- * multiplication
- / division

Parentheses can be used to determine the order in which the operations are performed. The computer does the operation in the innermost parentheses first and works its way out. These operations can be used in a variety of ways, such as in the following programs:

```
10 INPUT X
20 A = (6 + 4 * (X - 3)) / 5
30 PRINT A

10 PRINT "INPUT 2 NUMBERS"
20 INPUT X, Y
30 PRINT X; "+"; Y; "="; X + Y
40 PRINT
50 PRINT X; "-"; Y; "="; X - Y
```

```
60 PRINT
70 PRINT X; "*" ; Y; "=" ; X * Y
80 PRINT X; "/" ; Y; "=" ; X / Y
90 PRINT
100 GOTO 10
```

ARRAYS

An array is an ordered list of numbers. To store an array in the computer's memory, you must first tell the computer how many numbers or "elements" there are in the list. This is done with a DIM statement. For example, DIM A(5) means that the array denoted by A has six elements A(0) through A(5). An array name is the same as the name of a numeric variable. It must begin with a letter and can contain only capital letters and numbers. A particular element in an array is referred to by giving the array name followed by the number of the element in parentheses. For example, A(4) refers to the fifth element of the array named A. The following program will store a pricelist in an array format. Here is the pricelist:

Item #	Price
1	\$1.75
2	2.50
3	6.25
4	1.25
5	3.10

```
10 DIM A(5)
20 A(1) = 1.75: A(2) = 2.50
30 A(3) = 6.25: A(4) = 1.25
40 A(5) = 3.10
50 PRINT "WHICH PRICE DO YOU WANT? "
60 INPUT N
70 PRINT
80 PRINT "PRICE "; N; " IS $"; A(N)
```

DIM

The DIM (dimension) statement is used to reserve enough space in the computer's memory to store an array. Every array (of more than 10 elements) must be preceded by a DIM statement. Here is an example:

```
10 DIM A(20)
```



```
20 FOR I = 0 TO 20
30 A(I) = I
40 PRINT A(I)
50 NEXT I
```

END

The END statement causes the computer to end its execution of a program. Here is an example. (Also try running this program without the END statement.)

```
10 PRINT "I 'M TIRED"
20 PRINT "I WANT THIS PROGRAM TO END"
30 END
40 GOTO 10
```

EQUALITY

In Commodore BASIC, the equality relations are denoted as follows:

- = equals
- > greater than
- >= greater than or equal to
- < less than
- <= less than or equal to
- <> not equal to

Here is an example:

```
10 INPUT A, B
20 IF A > B THEN PRINT A; ">"; B
30 IF A <> B THEN PRINT A; "<>"; B
40 IF A = B THEN PRINT A; "="; B
50 IF A < B THEN PRINT A; "<"; B
```

In the case of string variables, these relations are the same as alphabetical order. Here is an example:

```
20 INPUT A$, B$
30 IF A$ > B$ THEN PRINT A$; ">"; B$
40 IF A$ <> B$ THEN PRINT A$; "<>"; B$
50 IF A$ = B$ THEN PRINT A$; "="; B$
60 IF A$ < B$ THEN PRINT A$; "<"; B$
```

FOR/NEXT

The FOR/NEXT statements tell the computer to repeat a sequence of instructions a specific number of times. Repeating a sequence of instructions is called a *loop*. For example, if you want the computer to print "SORRY FOR THE DELAY" 27 times, you could use the following program:

```
10 FOR N = 1 TO 27
20 PRINT N; " SORRY FOR THE DELAY. "
30 NEXT N
```

Each time the computer reaches the NEXT N statement the value of N is increased by 1 and the computer goes back to the FOR statement. When the value of N reaches the number specified in the FOR statement (in this case 27), the computer is finished with the loop and proceeds to the statement after the NEXT statement.

If you want the computer to count in increments other than 1 in the FOR/NEXT loop you can follow FOR with a STEP instruction. For example, the instruction

```
FOR N = 4 TO 30 STEP 2
```

will have the computer start the loop with N = 4 and

increase the value of N by 2 each time the loop is completed until N= 30. The following programs illustrate the use of the STEP instruction:

```
10 FOR K = 1 TO 9 STEP 2
20 PRINT K
30 NEXT K
```

```
10 FOR J = 3 TO 24 STEP 3
20 PRINT J
30 NEXT J
```

```
10 FOR L = 15 TO 0 STEP -1
20 PRINT L
30 NEXT L
```

By having no instructions between the FOR and NEXT statements you can cause a time delay. This is because the computer will run through the loop the specified number of times without executing any instructions. The following program illustrates this. You can experiment by changing the ending value of K from 100 to 200 to 500 and then leaving out line 20.

```
10 PRINT "HELLO";
20 FOR K = 1 TO 100: NEXT K
30 GOTO 10
```

GOSUB and RETURN

The GOSUB and RETURN statements instruct the computer to go to a certain set of instructions, called a subroutine, and then return to the program once this set of instructions is performed. The GOSUB instruction sends the computer to the subroutine and the RETURN statement causes the computer to return to the line following the GOSUB statement. Subroutines are useful if you want the computer to perform the same set of instructions at different times during the course of a program. Here is a program that illustrates the use of subroutines:

```
10 A = 6: B = 10: C = 9
20 GOSUB 100
30 A = 8: B = 9
40 GOSUB 100
50 A = 10: B = 2: C = 1
60 GOSUB 100
70 END
100 PRINT "A="; A; " B="; B; " C="; C
110 RETURN
```

GOTO

The GOTO statement instructs the computer to go to a specified line number in a program for its next instruction. Here is an example:

```
10 PRINT "I WILL NOT TALK IN CLASS"  
20 FOR K = 1 TO 50: NEXT K  
30 GOTO 10
```

GRAPHICS

Memory location 53280 determines the border color and 53281 determines the background color of the screen. These locations accept numbers from 0 to 15. The following program illustrates the different background and border colors in a "forest" setting:

```
10 FOR I = 1 to 999  
20 PRINT " SHIFT A";  
30 NEXT I  
40 FOR B = 0 TO 15  
50 FOR C = 0 TO 15  
60 POKE 53280, B
```

```
70 POKE 53281, C
80 FOR M = 1 TO 2000; NEXT M
90 NEXT C
100 NEXT B
110 GOTO 110
```

The following table shows the different colors available on the Commodore 64:

0 BLACK	8 ORANGE
1 WHITE	9 BROWN
2 RED	10 LIGHT RED
3 CYAN	11 GRAY 1
4 PURPLE	12 GRAY 2
5 GREEN	13 LIGHT GREEN
6 BLUE	14 LIGHT BLUE
7 YELLOW	15 GRAY 3

The computer's screen is a grid of 1000 "characters" made up of 25 rows and 40 columns. The computer's memory stores information about these 1000 screen location in a screen memory and a color memory. The screen memory begins at memory location 1024 and ends at location 2023. Location 1024 is the upper left

corner of the screen. Location 1025 is the location of the next character to the right of that, location 1063 is the rightmost position of the first row, location 1064 is the location of the first character in the second row, etc. To put a character at a particular location use the POKE $1024 + M + 40 * N, C$ where M is the column, N is the row, and C is the "screen display code." The characters are defined by their screen display codes, which range from 0 to 255. (Codes 128 to 255 are reversed images of codes 0 to 127.) To see different characters with codes 0 to 127 displayed in column 20, row 12, RUN the following program. For more information on the screen display codes, consult your *Commodore 64 User's Guide*.

```
10 FOR I = 0 TO 127
20 POKE 1024 + 20 + 40*12, I
30 FOR J = 1 TO 100: NEXT J
40 NEXT I
```

The color memory is contained in the 1000 locations beginning at 55296 and ending at 56295. You can POKE any of the 16 Commodore colors into a particular location and have a character POKED onto the screen appear in that color. To see this, RUN the following program.

For more information on the "Color memory map," consult the *Commodore 64 User's Guide*.

```
10 FOR I = 0 TO 15
20 POKE 55296 + 20 + 40*12, I
30 POKE 1024 + 20 + 40*12, 105
40 FOR J = 1 TO 100: NEXT J
50 NEXT I
```

IF-THEN

The IF-THEN statement instructs the computer to perform an instruction if a certain condition is met. If the condition is not met the computer goes to the next line in the program. Here is an example:

```
10 INPUT X
20 IF X >= 0 THEN GOTO 40
30 PRINT "X IS NEGATIVE": GOTO 10
40 PRINT "X IS NONNEGATIVE": GOTO 10
```

INPUT

The INPUT statement instructs the computer to receive information entered from the keyboard. The information will be stored in the computer either as a numeric or a string variable. The following programs illustrate the INPUT statement:

```
10 INPUT X
20 PRINT
30 PRINT "THE VALUE OF X EQUALS "; X
```

```
10 INPUT A$
20 PRINT
30 PRINT "HERE IS A$: "; A$
```

INT(X)

INT(X) instructs the computer to drop the fractional part (if there is one) of the number X. Here are some examples of how this function works:

```
INT (3.14) = 3
```

```
INT(.7581) = 0
INT(2) = 2
INT(1758.9) = 1758
```

Here is a program that picks random integers between 1 and 6. This program simulates the roll of a die. First we use the RND(1) function to pick a random decimal fraction. Multiplying by 6 and adding 1 will give us a random number between 1 and 7. Using INT(X) to drop the fractional part will give us an integer between 1 and 6, the simulated roll of a die.

```
10 X = INT(6 * RND(1)) + 1
20 PRINT X;
30 FOR L = 1 TO 50: NEXT L
40 GOTO 10
```

LET

The LET statement instructs the computer to store information as a numeric or string variable. It is not necessary to use the word LET in the instruction. Here are two examples:

```
10 LET X = 20
```

```
20 PRINT X + 10
20 A$ = "FROGS"
30 PRINT "I LIKE "; A$
```

LIST

Typing LIST and pressing the **RETURN** key causes the computer to list the program currently in its memory.

NEW

Typing NEW and pressing the **RETURN** key causes the computer to clear its memory of any program so you can start writing a new program.

POKE

The "POKE" command enables you to put values into specific locations in the computer's memory. In Commodore BASIC, the POKE command is needed to create sound and certain graphics effects.

PRINT

The PRINT statement instructs the computer to print a message on the screen. You must put the message to be printed in quotes as follows:

```
PRINT "WHAT'S IN A GAME? "
```

You can also print information stored in a variable, as in the following examples:

```
10 A$ = "HELP"  
20 PRINT A$
```

```
10 A = 75  
20 PRINT A
```

You can do arithmetic with a PRINT statement without writing a program. Here are some examples:

```
PRINT 9 + 2  
PRINT 6 * 7  
PRINT 8 / 2
```

Commas and semicolons can be used to control the space after PRINT statements.

No punctuation after a PRINT statement causes the

computer to skip to the next line before printing anything else.

A comma after a PRINT statement causes the computer to skip to the next "TAB" position before printing anything else. Commodore tab settings are every 10 spaces across the line. If you are within 10 spaces of the end of a line, the computer will skip to the next line if the PRINT statement is followed by a comma.

A semicolon after a PRINT statement will cause the computer to start printing again in the space after it stopped printing. Here is an example:

```
10 PRINT
30 PRINT "THIS SENTENCE ";
40 PRINT "IS FALSE"
```

READ and DATA

The READ statement instructs the computer to read from a list of information in the program. This information follows a DATA statement. The information can be read

in either numeric or string variable form. The READ and DATA statements allow you to store information in a program that can be used in the course of the program. The READ statement can be used repeatedly in the program and will read information from the DATA statement or statements in the order in which it appears. Here is an example that illustrates the READ and DATA statements:

```
20 READ A, B
30 PRINT A, B
40 READ A$, B$
50 PRINT "PROGRAMMING IS "; A$; B$
60 DATA 8, 15, FUN
70 DATA (SOMETIMES)
```

RESTORE

The RESTORE statement sends the computer to the beginning of a DATA list to use the same data more than once. Here is an example:

```
20 READ A$, B$
30 PRINT A$; " ARE FUN"
```

```
40 FOR V = 1 TO 500: NEXT V
50 PRINT B$; " ARE DULL "
60 FOR V = 1 TO 500: NEXT V
70 RESTORE
80 GOTO 20
100 DATA MACHINES, PEOPLE
```

RND(1)

RND(1) instructs the computer to select a decimal fraction at random. Here is an example that prints a sequence of random decimal fractions:

```
10 PRINT RND (1)
20 FOR V = 1 TO 100: NEXT V
30 GOTO 10
```

RUN

Typing RUN and pressing the **RETURN** key causes the computer to run the program currently in its memory.

SOUND

Sound is controlled by memory locations 54272 through 54296. Locations 54296, 54273, 54272, and 54276 control the volume, high and low frequency pitches, and the waveform for VOICE 1. (There are three voices all together.) For example, the volume settings range from 0 (no sound) to 15 (loudest). To see these effects, RUN the following program. For more information consult the *Commodore 64 User's Guide*.

```
10 FOR L = 54272 TO 54296: POKE L, 0: NEXT L
20 V = 54296: H = 54273: L = 54272: W = 54276
30 S = 54278: A = 54277
40 POKE V, 15
45 FOR P = 1 TO 255
50 POKE H, P: POKE L, 37
60 POKE S, 240
70 POKE W, 17
75 NEXT P
80 POKE W, 16
```

TAB

The TAB function is used in the PRINT statement. TAB(X) will cause the next item to be printed in space X on the line. (A line has 40 spaces). Here is an example:

```
10 FOR I = 1 TO 20
20 PRINT I; TAB (10+I) ; 20-I
30 NEXT I
```

VARIABLES

A variable is a device for storing information in the computer's memory. It is like a box with a label. This label or variable name must begin with a letter and must contain only capital letters and numbers. A variable that will be used to store only numbers is called a *numeric variable*. A variable that will represent non-numeric information is called a *string variable* and must end with the dollar sign symbol \$. Here are two examples:

```
20 LET A$ = "FRANK"
30 PRINT "WHO IS "; A$; "?"

10 INPUT X, Y
20 PRINT
30 PRINT X; "+ "; Y; "= "; X + Y
```

INDEX OF PROGRAMMING TERMS

(Page numbers indicate
introduction of the term)

Arithmetic Statements, 25	TAB, 11, 73
Arrays, 45	VARIABLES, 12, 26
DIM, 46	
END, 58	
Equality Relations, 37	
FOR/NEXT, 9	
FOR/NEXT/STEP, 68	
GOSUB-RETURN, 57	
GOTO, 3	
GRAPHICS characters, 52	
IF-THEN, 35	
INPUT, 13	
INT(X), 34	
LET, 35	
LIST, 3	
NEW, 3	
POKE, 50, 117	
PRINT, 2	
READ-DATA, 70	
RESTORE, 71	
RND(1), 34	
RUN, 3	
SOUND, 49, 104	

\$12.95

Michael Orkin Ed Bogas

SURVIVAL on PLANET X

with the Commodore 64®

Follow the space adventures of Vivian, the teenage space explorer on your Commodore 64. While traveling through space in her spaceship, Vivian answers a call for help from Planet X. When she reaches the planet's surface, she finds herself in many adventure-filled situations and **you** can help her out! As you read about Vivian's adventures on Planet X, you'll also learn how to program in BASIC. All the programs are about Vivian, the planet, and the lizard people that inhabit Planet X. By entering the programs on your Commodore 64, you'll see: how Vivian escapes from jail after the Leadership Council puts her there; how she meets Xaviar, the Rebel Leader; the ride on the trolley to find her spaceship; how time is measured on Planet X; and how Vivian and her new-found friends escape from the army lizards. As designers of video games, the authors add arcade-like excitement to the basic ideas and techniques of programming. Not only will you learn about programming in BASIC but you'll discover some of the tricks and techniques that are used in your favorite arcade games!

About the authors:

Michael Orkin is a video game designer and professor of statistics in California. Ed Bogas has also done video game designing and is a composer and producer in the entertainment business.

RESTON PUBLISHING COMPANY, INC.
A Prentice-Hall Company
Reston, Virginia



0-8359-7413-8